

**INDUSTRIALES Y DE TELECOMUNICACIÓN**

**UNIVERSIDAD DE CANTABRIA**



**Trabajo Fin de Grado**

**Estudio de aplicación de tecnología  
IBeacon para el posicionamiento en  
interiores**

**(Application study of IBeacon technology  
for indoor positioning)**

**Para acceder al Título de**

**Graduado en Ingeniería de Tecnologías de  
Telecomunicación**

**Autor: Ángel Herrero Monge**

**Septiembre-2021**



# Índice

---

<b>Resumen</b> .....	1
<b>Abstract</b> .....	2
<b>1 INTRODUCCIÓN Y OBJETIVOS</b> .....	3
1.1 Introducción .....	3
1.2 Objetivos y Motivación del trabajo .....	4
1.3 Organización del documento .....	6
<b>2 MARCO TEÓRICO</b> .....	7
2.1 Localización en interiores .....	7
2.2 Tecnología Bluetooth .....	10
2.3 Tecnología IBeacon .....	13
2.4 Cálculo de la posición: Trilateración .....	18
<b>3 ENTORNO DE DESARROLLO</b> .....	23
3.1 Elementos físicos .....	23
3.2 Elementos Software .....	26
<b>4 ALGORITMO DE LOCALIZACION: Propuesta inicial y Nueva propuesta</b> .....	32
4.1 Escenario empleado en ambas propuestas .....	32
4.2 Resultados de la propuesta anterior .....	34
4.3 Nueva propuesta del algoritmo de localización en interiores .....	36
<b>5 CONCLUSIONES Y POSIBLES MEJORAS</b> .....	55
5.1 Conclusiones .....	55
5.2 Posibles mejoras en el futuro .....	56
<b>6 BIBLIOGRAFÍA</b> .....	57

# Índice de Figuras

---

Figura 1 – Origen del logo Bluetooth .....	10
Figura 2 – Estructura de una piconet .....	12
Figura 3 – Estructura de una Scatternet .....	12
Figura 4 – Formato de la trama Bluetooth .....	14
Figura 5 – Formato del paquete IBeacon .....	14
Figura 6 – Ranging de los IBeacon .....	17
Figura 7 – Representación de la trilateración 2D .....	18
Figura 8 – Representación de la trilateración 3D .....	19
Figura 9 – Planteamiento geométrico de la trilateración .....	20
Figura 10 – Posibles casos de trilateración .....	20
Figura 11 – Puntos del triángulo .....	22
Figura 12 – Módulo DSD TECHHM10 .....	23
Figura 13 – Batería LiPo de 500mAh/3.7V .....	25
Figura 14 – TracerPlus Software .....	26
Figura 15 – Interfaz de usuario de la app de Android .....	27
Figura 16 – Interfaz del IBeacon Scanner .....	29
Figura 17 – Interfaz de Microsoft Excel .....	30
Figura 18 – Escenario del trabajo anterior .....	33
Figura 19 – Escenario del trabajo actual .....	33
Figura 20 – Resultados del primer escenario .....	34
Figura 21 – Resultados del segundo escenario .....	35
Figura 22 – Resultados del tercer escenario .....	35
Figura 23 – Fichero resultante de agrupar todas las medidas .....	38
Figura 24 – Histograma de los errores promedio .....	38
Figura 25 – Histograma de los errores promedio .....	39
Figura 26 – Datos reordenados para la trilateración .....	43
Figura 27 – Ejemplo de la nube de puntos obtenida .....	49
Figura 28 – Nubes de puntos/Método Path Loss .....	51
Figura 29 – Nubes de puntos/Método Ranging .....	52

# Índice de Fórmulas

---

Fórmula F.1 – Estimador RSSI .....	16
Fórmula F.2 – Estimador de “n” .....	16
Fórmula F.3 – Estimador de la distancia .....	17
Fórmula F.4 – Matriz “DIST” .....	44
Fórmula F.5 – Trilateración .....	45

# Índice de Tablas

---

Tabla 1 – Clases de dispositivos Bluetooth .....	11
Tabla 2 – Errores promedio .....	38
Tabla 3 – Errores promedio en función de la orientación .....	39
Tabla 4 – Resultados obtenidos método Path Loss .....	53
Tabla 5 – Resultados obtenidos método Ranging .....	53

# Resumen

El propósito principal de este trabajo es analizar y desarrollar el sistema de posicionamiento en interiores mediante la inclusión de la tecnología IBeacon. La tarea principal es resolver el problema de posicionamiento en interiores a través del planteamiento de un nuevo sistema de balizas utilizando módulos Bluetooth.

El trabajo parte del TFG de Carlos Calzada donde se estudiaba la inclusión de estos módulos en un sistema de balizas que permitiese obtener la posición del usuario a partir de la potencia recibida de cada módulo para aplicar sobre el método de trilateración. Los resultados, a pesar de ser bastante decentes, arrojaban un error medio un tanto ineficiente en los tres escenarios planteados, lo que permitía a las líneas futuras reforzar su trabajo incluyendo una serie de modificaciones.

En base a los resultados obtenidos, se ha integrado en el sistema diseñado por Carlos la inclusión de la tecnología Ibeacon en los módulos Bluetooth empleados como balizas. Esta incorporación permite un estudio del posicionamiento en interiores con alguna serie de novedades, entre las cuales se incluye modos de ahorro de energía en las balizas y la obtención directa de la distancia sin necesidad de hacer cálculos mediante un parámetro denominado “ranging”. Esto último es esencial en el estudio de este trabajo, ya que permite comparar los resultados de posicionamiento empleando el método de hallar la posición utilizando la potencia recibida y mediante esta nueva tecnología. Por último, dado que era una de las posibles conclusiones en el trabajo anterior, se ha estudiado mediante pruebas reales el error de posicionamiento del dispositivo móvil en base a la posición relativa del usuario, es decir, si se sitúa de frente, de espaldas o perpendicularmente a los módulos.

El sistema resultante determina un error entre la medida real y la obtenida de pocos metros, siendo bastante ineficiente, pero permitiendo aportar nuevas soluciones en el futuro al sistema planteado mediante posibles líneas futuras que serán comentadas en el apartado de conclusiones.

# Abstract

The main purpose of this work is to analyze and develop the indoor positioning system by including IBeacon technology. The main task is to solve the indoor positioning problem through the approach of a new beacon system using Bluetooth modules.

The work starts from Carlos Calzada's TFG where the inclusion of these modules in a beacon system was studied that allows obtaining the user's position from the power received from each module to apply on the trilateration method. The results, despite being quite decent, yielded a somewhat inefficient mean error in the three scenarios proposed, which allowed future lines to reinforce their work by including a series of modifications.

Based on the results obtained, the inclusion of Ibeacon technology in the Bluetooth modules used as beacons has been integrated into the system designed by Carlos. This incorporation allows a study of indoor positioning with some series of novelties, among which are included energy saving modes in the beacons and the direct obtaining of the distance without the need to make calculations by means of a parameter called "range". The latter is essential in the study of this work, since it allows comparing the positioning results using the method of finding the position using the received power and using this new technology. Finally, since it was one of the possible conclusions in the previous work, real tests have been studied of the positioning error of the mobile device based on the relative position of the user, that is, if it is located from the front, from the back or perpendicularly to modules.

The system resulted in a certain error between the real measurement and that obtained of a few meters, being quite inefficient, but allowing new solutions to be provided in the future to the system proposed through possible future ones that will be commented on in the conclusions section.



# 1. INTRODUCCIÓN Y OBJETIVOS

Este primer apartado sirve como breve introducción al problema que se propone, así como la solución abordada, los puntos en los que se basa y, por último, la estructura que sigue este documento.

## 1.1 Introducción

Los sistemas de localización suponen una de las manifestaciones más populares del desarrollo actual de las IT, experimentando un considerable aumento de popularidad conforme pueden ser aplicados a soluciones de movilidad, que son las más demandadas en estos momentos.

El propósito principal de los sistemas de localización es determinar la ubicación física de un objeto o una persona en un entorno físico con relación a un sistema de coordenadas para, en la mayoría de los casos, procesar y acceder a una información específica, relacionada con dicha posición. Los sistemas de posicionamiento tradicionales están basados en el uso de información extraída de satélites, como es el caso de GPS (Global Positioning System), permiten indicar la ubicación geográfica de un usuario con un error mínimo de 3 a 10 metros, condicionado a la recepción sin obstáculos de las señales satelitales, y al uso de métodos adicionales de corrección, con sistemas base terrestres.

Cuando se habla de posicionamiento en el interior de un edificio, la señal GPS queda condicionada, en el mejor de los casos, por los efectos de la atenuación y multicamino de las referencias satelitales/terrestres. Por esta razón, se precisan nuevas soluciones que permitan trabajar sin problemas en espacios interiores, aunque ya existen experiencias previas que se basan, por ejemplo, en sensores de ultrasonidos o de infrarrojos. A pesar del éxito de estos sistemas, supone un gran inconveniente a nivel de escalabilidad para aplicarlo a mayores, ya que se precisa el despliegue de sensores específicos en cada uno de los recintos en los que se quisiera aplicar la localización, y muchas veces dicho posicionamiento solo permite asegurar áreas probables, no posiciones reales.

El nivel de implicación en el desarrollo de soluciones para lograr la equidad entre la localización en exteriores e interiores ha aumentado exponencialmente en los últimos años debidos, especialmente, a la gran inversión de tiempo y capital por parte de gigantes como Google o Samsung, entre otros. Algunas de estas soluciones

comerciales pasan precisamente por la necesidad de instalar una infraestructura dedicada, por lo que no son aplicables de forma generalizada y tienen grandes problemas cuando son aplicadas en escenarios a mayor escala.

Como principal mecanismo para la determinación de posiciones caben destacar como precursor [3], El **Wifi fingerprinting**. De los primeros en ser empleados, aprovecha la medición de la potencia de las diferentes señales Wifi, de forma que registra dichas medidas a lo largo de diferentes posiciones de las estancias, y obtiene un mapa de referencia. Cuando un usuario quiere determinar su posición, realiza una medida en su dispositivo móvil y busca una coincidencia con cada punto del mapa de referencia. La evolución de las tecnologías wireless ha generalizado este mecanismo como **fingerprinting**, en general, ya que actualmente se emplean todo tipo de señales captadas por el móvil para realizar el mapa de referencia (Bluetooth, NFC, QRcodes, etc, no sólo Wifi). Como caso especial, se encuentran las técnicas de análisis de **Campo magnético**, en las que se emplean las mediciones de la variación del campo magnético terrestre, producidas por el terreno y las edificaciones para realizar un fingerprinting basado en líneas en vez de puntos, siendo una técnica más útil para el posicionamiento que las dos anteriores. Sin embargo, su aplicación en los dispositivos móviles de usuario es bastante limitada.

En cualquier caso, el principal problema de la mayoría de los sistemas existentes radica en el coste y esfuerzo de implementar, ya sea estructuras dedicadas, u obtener la cartografía de fingerprinting, además de la poca precisión que permiten ofrecer estos métodos fuera de los estudios en laboratorio. Las medidas utilizadas en los diferentes métodos, fluctúan enormemente por las características propias de las señales y su propagación, el umbral de detección de los sensores, la radiación absorbida y emitida por las personas y el propio usuario, interferencias, etc. No es de extrañar que, en los últimos tiempos, se esté buscando de forma desesperada alternativas mucho más simples y, sobre todo, más baratas.

## 1.2 Objetivos y Motivación del trabajo

El principal objetivo de este trabajo es analizar alternativas a los métodos existentes para el posicionamiento en interiores, bajo la premisa de minimizar costes y obtener precisiones aceptables, además de que sea lo suficientemente sencilla como para que se pueda proporcionar al usuario una solución sencilla que pueda utilizar en su vida cotidiana como si de una aplicación más mainstream como Google Maps se tratara.

La idea de este trabajo es continuista, puesto que retoma los resultados de un Trabajo Fin de Master anterior, de Carlos Calzada Cabano [1], así como de otros TFM y Trabajos Fin de Grado anteriores. El trabajo utilizado como referencia hace uso de la tecnología Bluetooth, al ser una tecnología altamente desarrollada y conocida por todos, con un empleo relativamente sencillo y que todos los dispositivos móviles integran en su sistema. El sistema desarrollado en [1], utiliza las medidas de potencia obtenidas directamente por el dispositivo móvil, de la señal Bluetooth, correspondiente a los

mensajes Beacon lanzados por otros dispositivos Bluetooth, dispuestos estratégicamente a modo de balizas. En este trabajo se estudia el uso de la tecnología IBeacon y aprovechar sus novedades y ventajas frente a la tecnología Beacon convencional. Para ello se ha partido de los siguientes objetivos:

- Desarrollar e implementar una alternativa a la solución de localización desarrollada en [1], aprovechando las características de la tecnología IBeacon.
- Comprobar la viabilidad del método sugerido, mediante pruebas reales del sistema que permitan comparar los resultados obtenidos empleando las medidas de Ibeacons frente a las medidas de Beacons convencionales.

Todo ello se plantea sin perjuicio de mantener las condiciones de sencillez y bajo coste asociadas a este tipo de soluciones, utilizando elementos fáciles de obtener y realizando un despliegue en un escenario real, replicable fácilmente tanto para la implementación de este sistema, como el desarrollo de nuevas versiones o soluciones.

## 1.3 Organización del documento

Tras este primer capítulo introductorio, en el segundo capítulo se explica el **Marco Teórico sobre** el que se basa este trabajo, incluyendo todos los conceptos fundamentales, incluyendo protocolos, especificaciones o métodos matemáticos relacionados.

En el siguiente capítulo, el tercero, se expone el **Entorno de Desarrollo** empleado para la aplicación de la solución propuesta. En este caso se incluye tanto el entorno físico, como el software utilizado, haciendo hincapié en los aspectos y funcionalidades más relevantes.

Posteriormente, en el capítulo 4, se plantea el **Algoritmo de Localización** utilizado, tanto su implementación original, como la propuesta de modificaciones que permiten mejorar sus resultados mediante la inclusión de la nueva tecnología asociada a Bluetooth. Además del algoritmo de cálculo de posición, se describe el escenario de experimentación, la distribución de las balizas elegidas, el desarrollo de las medidas, y el análisis de los resultados obtenidos.

En el último capítulo se exponen y comentan las principales **conclusiones** relacionadas a los resultados obtenidos, así como posibles propuestas de mejora de cara futuros desarrollos.

## 2. MARCO TEÓRICO

En este capítulo se definen y exponen los conceptos teóricos y matemáticos empleados en el desarrollo del nuevo sistema de posicionamiento en interiores que se plantea en este trabajo.

### 2.1 Localización en interiores

Los sistemas de posicionamiento son procedimientos, normalmente automatizados, que permiten detectar la posición de un objeto en un espacio físico mediante la recopilación de información del entorno mediante una combinación de elementos hardware y software. El problema de posicionar cualquier elemento físico en un espacio abierto ha sido estudiado ampliamente, y ha permitido desarrollar sistemas con la capacidad tecnológica de llevar a cabo la tarea con errores bastante razonables. Dentro de este modelo de espacio abierto se distinguen dos sistemas de localización dependiendo de su alcance: globales y locales.

- **Los sistemas de posicionamiento globales o GPS** han supuesto uno de los avances tecnológicos más importantes en las últimas décadas y consiste en un sistema compuesto por una red de satélites (normalmente 24) situados en órbita alrededor de la Tierra, que envían señales que permiten calcular la distancia al usuario y así poder calcular su posición, normalmente mediante triangulación y/o trilateración. Esta tecnología estuvo exclusivamente destinada para fines militares hasta 1980, cuando el gobierno de Estados Unidos permitió su uso civil. Cabe mencionar que el GPS funciona las 24 horas del día de forma gratuita y en cualquier lugar del mundo sin importar la condición climática.
- **Los sistemas de posicionamiento locales o LPS** hacen uso de tecnología e infraestructura local, como puntos de acceso Wifi o antenas de telefonía, para lograr obtener la posición de un objeto físico en un espacio abierto, y solamente dentro de ámbitos geográficos muy bien definidos.

A pesar de que estos sistemas son bastante precisos a la hora de localizar objetos en un espacio exterior, en espacios interiores, donde no existe visión directa satelital, se producen grandes márgenes de forma errática y, es por ello, que surgen diferentes sistemas alternativos para aplicar a este tipo de entornos: **Los sistemas de posicionamiento en interiores o IPS.**

Estos sistemas surgen como un caso particular de los LPS, aplicables a entornos cerrados, con un mismo objetivo, pero distintas particularidades, tanto a nivel metódico, como tecnológico. Dentro de las tecnologías empleadas para estos sistemas, no existe un estándar impuesto, y han surgido diferentes alternativas válidas basadas en el uso de óptica, radio, u ondas magnéticas o acústicas. Actualmente la mayoría de los sistemas

indoor hacen uso de tecnologías radio, distinguiendo según el uso de ondas de banda estrecha (Bluetooth, Wifi) y ondas de banda ancha (UWB).

Si bien los sistemas anteriores pueden llegar a resultar bastante útiles y posicionar con bastante precisión, el principal problema del posicionamiento en interiores radica en la escasa accesibilidad por parte de la población a este tipo de tecnologías, al menos de una manera cotidiana y sencilla, como puede ocurrir con la localización en exteriores.

### *2.1.1 Sistemas Indoor existentes*

A pesar de lo comentado anteriormente, cabe mencionar la existencia de algunas soluciones implementadas y usadas actualmente a través de dispositivos cotidianos como un teléfono móvil, siendo lo que interesa realmente en este trabajo: proporcionar una solución barata, sencilla y escalable al problema de posicionamiento en interiores.

- **Triangular Wifi:** Propuesta por Google, esta tecnología [4] promete aprovechar diferentes puntos de acceso Wifi abiertos en una determinada zona, así como sus nodos, para ubicar al usuario con bastante precisión, sin recurrir a ninguna infraestructura ni tecnología nueva. Su principal ventaja es la capacidad de realizar el posicionamiento en interiores sirviéndose únicamente de un dispositivo móvil con conexión a internet, por lo que sería totalmente accesible a la mayor parte de la población sin recurrir en gastos adicionales. El problema radica en la escasa existencia de puntos wifi accesibles para cualquier usuario desde un entorno aleatorio, ya que la mayoría suelen tener algún tipo de cifrado, y más si se trata de puntos wifi domésticos situados en un bloque de edificios, por ejemplo.
- **Propuesta de Broadcom:** Esta empresa apuesta por el aprovechamiento de las herramientas internas del dispositivo móvil para realizar el posicionamiento en interiores sirviéndose de los datos proporcionados por el giroscopio o el acelerómetro para hallar la dirección y velocidad del usuario y, por tanto, ubicarlo en un entorno físico. Uno de los problemas principales de esta tecnología [4] es la necesidad de recurrir a otro sistema inicial que realice una primera ubicación, sirviendo como punto de partida para realizar los cálculos que permitan obtener la posición del usuario una vez se haya conseguido hallar su velocidad y dirección.
- **NFC:** Esta tecnología [4] se basa en señales de radiofrecuencia en donde se dirige un dispositivo móvil a dichas señales para identificarlo y localizarlo en un espacio reducido. Requiere implicación directa por parte del usuario, siendo esto su principal problema, la escasa autonomía por parte de la infraestructura para localizar el dispositivo móvil sin una implicación física directa.

- **Campo magnético:** Esta tecnología [3] apuesta por el aprovechamiento de las características físicas de los edificios, en concreto de su estructura y componentes metálicos (molduras, marcos, anclajes, etc.), así como de los elementos eléctricos que lo integran, formando perturbaciones en el campo magnético local. Estas perturbaciones permitirían realizar un mapa que sirviese como huella magnética con una finalidad similar al fingerprinting antes comentado. El principal problema de esta tecnología radica en la necesidad de realizar mapas constantemente en caso de cambio en la estructura o componentes que forman el entorno donde se ha medido el campo magnético, lo cual puede ser bastante frecuente (nuevo material, nuevas estanterías, recolocación de diversos componentes o nuevos cables, etc.). A esto hay que sumarle la necesidad de tomar una cantidad bastante elevada de medidas y moverse grandes distancias para realizar el primer mapa inicial para ubicar el campo magnético como punto inicial de partida. Esta desventaja hace que estos sistemas, a pesar de resultar interesantes y bastante precisos, se suelen utilizar en combinación a sistemas waypoint como los beacons.
- **Beacons:** Esta tecnología [2] es la escogida para el estudio de este trabajo. Se basa en la emisión de señales Bluetooth por parte de dispositivos autónomos BLE que pueden ser recogidas por cualquier dispositivo externo que tenga integrado un chip Bluetooth, como un Smartphone o una Tablet. Estas señales permiten triangular la posición del usuario con relativa facilidad en zonas concretas con un área reducido. Como principal desventaja encontramos errores al realizar el posicionamiento, obteniendo resultados con cierta imprecisión debido al escaso alcance de los beacons y la volatilidad en los resultados de las medidas. Pese a ello, esta tecnología está en constante evolución, y grandes empresas como Google con Eddystone o Apple con Ibeacon están aportando soluciones que añaden información adicional en los formatos originales. Entre los principales usos de estos métodos encontramos el acceso a información de interés basada en la localización del usuario en museos, la navegación en interiores de estaciones de metro o aeropuertos, y la gestión de inventario en plataformas IoT.

La solución por la que apuesta este trabajo es un sistema de localización en interiores basado en tecnología bluetooth mediante la emisión de beacons. Estas señales, emitidas por varias balizas colocadas en localizaciones concretas en un espacio con dimensiones conocidas, son capturadas por un dispositivo móvil que, con diversos cálculos, obtiene la posición relativa del usuario. Como principal novedad, se introduce la inclusión de la tecnología IBeacon, patentada por Apple para dicho propósito. Esto permite como principal novedad, determinar la posición del usuario evitando el cálculo de distancia antes de determinar la posición, a partir del uso de un parámetro exclusivo de esta tecnología denominado ranging.

## 2.2 Tecnología Bluetooth

Bluetooth es la tecnología elegida en varios de los proyectos, comentados con anterioridad, para la implementación del algoritmo de localización en interiores, dado su rendimiento, eficiencia y bajo coste. A través de esta tecnología los datos pueden viajar de manera eficaz sin necesidad de redes cableadas ni líneas de visión directa.

### 2.2.1 Introducción a la tecnología Bluetooth

El término Bluetooth se refiere a la tecnología de red desarrollada por el Grupo de Trabajo IEEE 802.15.1 del IEEE (Institute of Electrical and Electronic Engineers) como estándar de la industria para la conectividad inalámbrica. La tecnología Bluetooth se utiliza para transmitir datos y voz entre dispositivos mediante enlaces de radiofrecuencia. Este intercambio puede ser realizado punto a punto, orientados a la conexión o punto a punto entre dos dispositivos digitales. El objetivo principal de esta tecnología es reemplazar la conexión tradicional cableada siendo esto particularmente beneficioso para dispositivos móviles como teléfonos inteligentes y tabletas.

El origen del nombre Bluetooth ("diente azul" en inglés) proviene del rey de Dinamarca y Noruega Harald Blåtand (en inglés, Harald Bluetooth) quien logró unificar los pueblos de Noruega, Dinamarca y Suiza, contribuyendo a que sus miembros pudieran comunicarse entre ellos. Es por ello, que los desarrolladores de esta tecnología emplearon las runas escandinavas *hagall* y *berkana* (que representan la H y la B) para diseñar el logo, dotándolo de color azul debido a que se dice que al vikingo le gustaban mucho los arándanos.



Figura 1: Origen del logo de Bluetooth.

En 1998 la empresa escandinava Ericsson, en asociación con Nokia (de origen también escandinavo), IBM, Intel y Toshiba formaron el Bluetooth Special Interest Group (Bluetooth SIG), desarrollando una solución tecnológica propia y dando luz a esta tecnología, hasta entonces secreta. Bluetooth SIG es en la actualidad una organización sin ánimo de lucro que engloba a cerca de 33000 organizaciones, definiendo los estándares Bluetooth y promoviéndolo a través del desarrollo y producción de dispositivos que empleen esta tecnología. Algunos de los impulsores más importantes son Apple, Microsoft, Lenovo, Intel o Toshiba, entre otros.



### 2.2.2 Especificaciones Radio

Cada dispositivo debe contar con un microchip, llamado Transceptor, que permite transmitir y recibir en una banda ISM sin licencia (Las bandas ISM permiten ser utilizadas sin ningún tipo de autorización con fines industriales, científicos o médicos, así como en entornos domésticos) moviéndose en un rango de frecuencias entre los 2,4GHz y los 2,480 GHz, sujeta a variaciones de ancho de banda en determinados países. En este rango de frecuencias conviven multitud de tecnologías (Wifi, ZigBee...) por lo que se encuentra sometida a posibles interferencias. Cada aparato posee una dirección MAC única de 48 bits para poder ser identificado.

Los dispositivos que emplean este protocolo pueden transmitir y recibir voz y datos siempre y cuando se encuentre dentro de su alcance, permitiendo situarse a diferentes distancias e incluso diferentes estancias de un edificio si la potencia de transmisión lo permite.

Bluetooth soporta 2 tipos de enlaces: enlaces asíncronos sin conexión (ACL) con una tasa máxima de 721 kbps para el canal ascendente y 57.6 kbps para el descendente, y enlaces síncronos orientados a conexión (SCO) para audio y voz (hasta 3 canales SCO a 64kbps).

Actualmente los dispositivos más utilizados son del tipo Bluetooth de baja energía, también denominado Bluetooth LE (abreviado como BLE), Bluetooth ULP (Ultra Low Power) y Bluetooth Smart. Es una evolución de la tecnología digital de radio (inalámbrica) que permite la reducción de los consumos de energía, y como consecuencia, su uso en dispositivos mucho más simples y con alimentaciones basadas incluso en pilas de botón.

### 2.2.3 Clases Bluetooth

Con relación a la potencia de transmisión y su cobertura efectiva, podemos clasificar los dispositivos como Clase 1, Clase 2, Clase 3 y clase 4, según sea dicho nivel de potencia, permitiendo la compatibilidad entre distintas clases. Esta potencia de transmisión se sitúa entre 1 y 100mw, siendo típicamente de 2,5mW.

Clase	Potencia media	Alcance Máximo
Clase 1	100mW / 20dBm	100 metros
Clase 2	2.5mW / 4dBm	20 metros
Clase 3	1mW / 0dBm	1 metro
Clase 4	0.5mW / -3dBm	0.5 metros

*Tabla 1: Clases de dispositivos Bluetooth.*

### 2.2.4 Topologías Bluetooth

Una conexión Bluetooth puede iniciarse en cualquier dispositivo, que actúa como maestro, encargado de controlar la conexión y el intercambio de la información frente a los esclavos, que se limitan a transmitir información cuando lo indique el maestro.

Este modelo de conexión maestro-esclavo permite formar redes Bluetooth que pueden existir durante un tiempo indefinido hasta que el maestro cese la conexión. En la práctica, el proceso de conexión de dos o más dispositivos vía Bluetooth se conoce como emparejado o pairing.

Partiendo de este modelo, se definen dos topologías:

- **Piconet:** En esta topología se define un nodo central como maestro que puede interconectarse con un máximo de 7 dispositivos que actuarán como esclavos. El maestro se encargará de establecer las conexiones de emisión y recepción, además de escoger el hop para mantener el enlace y el salto de frecuencia

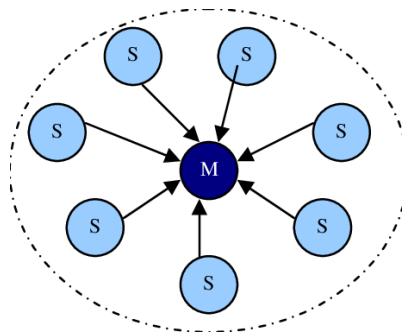


Figura 2: Estructura de una Piconet.

- **Scatternet:** Es un tipo de red ad hoc que consta de dos o más piconets solapados parcial o completamente. Un maestro de una piconet puede ser esclavo de otra, pero nunca maestro de dos redes.

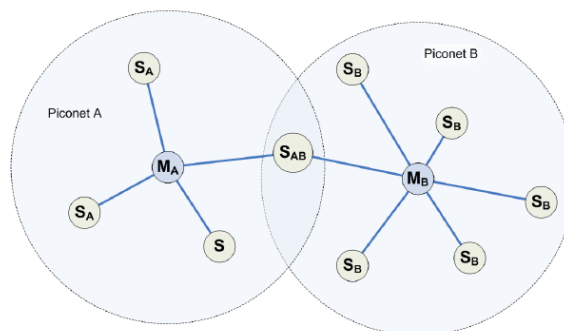


Figura 3: Estructura de una scatternet..

Pese a la existencia de estas topologías, su empleo no ha sido necesario para las pruebas prácticas ya que no se requiere enlazar el dispositivo móvil con las balizas BT, simplemente se necesita recoger la información de los beacons que envían dichos módulos para realizar los cálculos de posicionamiento necesarios. No obstante, se podría estudiar la posibilidad de emplear alguna de las anteriores tipologías de cara a la extracción de información adicional de las balizas en futuros trabajos.

## 2.3 Tecnología IBeacon

Como novedad en este trabajo y, como principal rasgo diferenciador de la línea de trabajos anteriores, se ha decidido introducir la tecnología IBeacon implementando este protocolo en las balizas BT para estudiar los cambios y/o posibles mejoras que se obtienen en la adquisición de los datos.

### 2.3.1 *Introducción a la tecnología IBeacon*

iBeacon [5] es la tecnología desarrollada por Apple para su sistema de posicionamiento indoor, basado en el uso de balizas Bluetooth de baja energía. Existen otras soluciones parecidas, aunque esta es la alternativa más implantada, hasta el punto de ser considerada casi como un estándar en la actualidad, con mayor cuota de mercado, incluso por encima de la opción denominada Eddystone, de Google.

Los dispositivos BLE con iBeacon son totalmente compatibles con casi cualquier dispositivo móvil con sistema operativo IOS o Android, así como para ordenadores que incluyan bluetooth 4.0 o superior.

### 2.3.2 *Uso y características*

Entre las principales características de BLE es que, en función de la clase utilizada, se obtiene un alcance teórico situado entre los 50 y 70 metros, sin perder la posibilidad de comunicar al dispositivo con teléfonos inteligentes. Esto significa que tiene que ser posible ubicar al usuario del móvil dentro del rango de acción de los dispositivos BLE utilizados, es decir, determinar, con alta probabilidad, que el usuario se encuentra en un determinado espacio, sin más que ver los mensajes de anuncio, denominados Beacon, que envían todos los dispositivos BLE circundantes, siempre que se conozca la posición de éstos.

Los Beacons son dispositivos que emplean BLE para transmitir a otros dispositivos que tengan compatibilidad y usen esta tecnología como Smartphones o Tablet. Su uso también permite detectar y reconocer dispositivos en sus inmediaciones con el fin de enviar algún tipo de mensaje. Actualmente su uso se basa principalmente en la navegación en interiores, los pagos sin contacto, fidelización y facturación automática.

En el caso de los iBeacons, estos mensajes de anuncio incluyen contenido adicional, originalmente pensados para uso comercial, para incluir, por ejemplo, recomendaciones personalizadas u ofertas basadas en el historial de compras del usuario sin la necesidad de su interacción directa, a diferencia de tecnologías como NFC, que requiere siempre que el usuario actúe , previamente.

### 2.3.3 Formato de los paquetes iBeacon

Bajo el formato BLE, un paquete iBeacon está formado por cuatro piezas fundamentales de información que componen el campo Data que está contenido en la PDU de la trama Bluetooth, mostrada a continuación en las siguientes figuras.

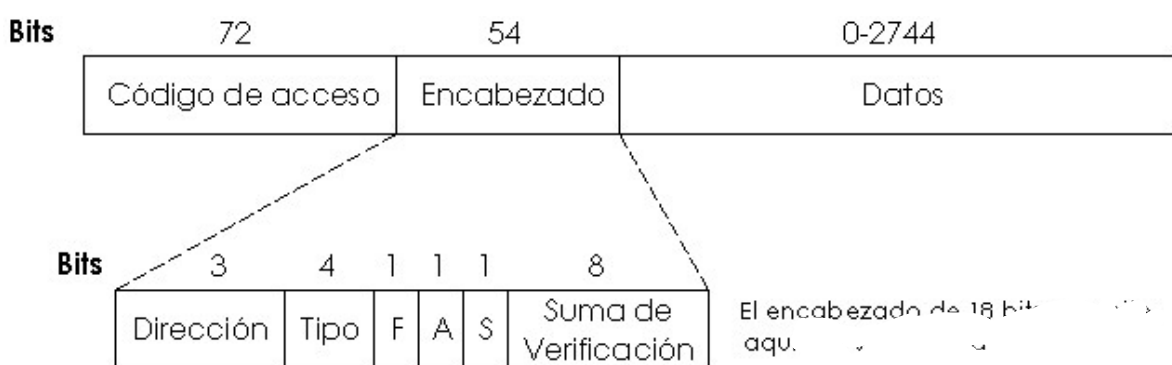


Figura 4: Formato de la trama Bluetooth.

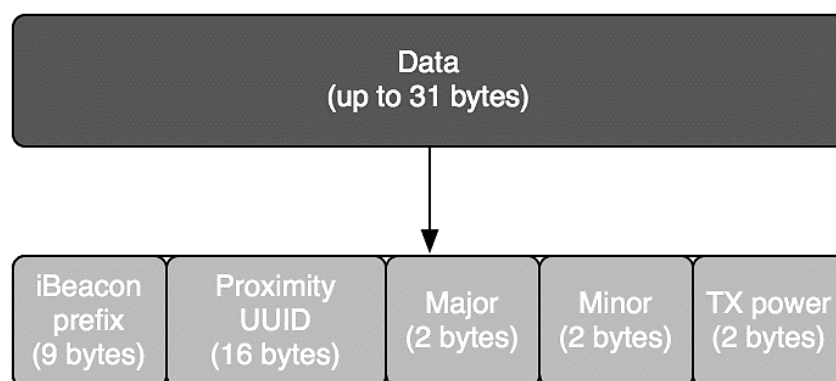


Figura 5: Formato de paquete iBeacon.

- **iBeacon prefix:** Contiene información del anunciante. Está compuesto por las flags, la cabecera, el tipo de iBeacon y su longitud, así como el ID de la compañía.
- **UUID:** Es el identificador principal y se trata básicamente de una cadena de 16 bytes que identifica cada grupo de balizas de forma única con el fin de distinguir Beacons de diferentes compañías en las inmediaciones.

- **Major y Minor:** Son dos valores Integer comprendidos entre 1 y 65535 bits. Major es una cadena de 2 bytes que se utiliza para distinguir un subconjunto de balizas dentro del conjunto de balizas UUID que lo contiene. Minor también está formado por una cadena de 2 bytes y se utiliza para identificar balizas individuales dentro del subconjunto formado en major.
- **Tx power:** Ligado a la función ranging, este parámetro se utiliza para determinar la distancia del usuario a la baliza. Emplea la intensidad de la señal a 1 metro del dispositivo para calibrar las balizas y con ello poder codificar los datos para obtener la distancia por adelantado. Los dispositivos móviles generalmente utilizan esto para dar una estimación aproximada de la distancia, como se pretende estudiar en este trabajo.

En el caso concreto de este trabajo, no se van a utilizar ninguno de estos valores más que TxPower, ya que los valores de UUID, Major y Minor no son diferenciadores para este estudio, pues nos vamos a guiar a través de la dirección MAC de los módulos para identificarlos, sin necesidad de realizar agrupaciones. El TxPower se empleará para el cálculo en la estimación de la distancia al dispositivo móvil, pero esto se hará de forma interna en la aplicación empleada para la toma de medidas, como se comentará en los próximos apartados.

## 2.4 Métodos para la obtención de la distancia

El algoritmo de posicionamiento empleado en este trabajo es denominado trilateración y, para la implementación del mismo, requiere conocer la distancia del dispositivo móvil desde el que se realizan las medidas hasta cada una de las balizas tomadas como referencia, al menos tres. Dicha distancia es fundamental para la aplicación del algoritmo en base al radio (la distancia al dispositivo móvil) formado por la circunferencia que se genera alrededor de cada baliza. Esto, gracias a la tecnología IBeacon, se puede realizar de dos maneras:

### 2.4.1 Distancia a partir de la RSSI

El indicador de fuerza de la señal recibida o en inglés RSSI (Received Signal Strength Indicator) es uno de los parámetros más interesantes para medir el nivel de potencia de la señal recibida por un dispositivo inalámbrico (en este caso mediante el protocolo Bluetooth) a través de una escala de referencia (comúnmente 1mW).

Empleando la siguiente fórmula se relaciona la distancia entre el módulo emisor y el

$$RSSI_i[dBm] = RSSI_0[dBm] - 10n \log_{10}\left(\frac{d_i}{d_0}\right) \quad (F.1)$$

módulo receptor de un beacon bluetooth con el parámetro RSSI [1]:

Donde:

- $RSSI_i$ : Potencia recibida estimada en  $dBm$ .
- $RSSI_0$ : RSSI tomado a una distancia de referencia  $d_0$  expresado en  $dBm$ .
- $n$ : Path loss o constante de pérdida de ruta, su valor típico esta entre 2 y 4.
- $d_i$ : distancia entre ambos módulos.
- $d_0$ : distancia de referencia, típicamente 1 o 2 metros.

El Path Loss depende del medio y los obstáculos existentes entre los dispositivos, pudiendo afectar a la propagación de la señal fenómenos como reflexión, difracción o dispersión debido a dichos obstáculos. En el espacio libre  $n=2$ .

Tomando la RSSI de referencia a una distancia  $d_0$  de 1 metro, se despeja  $n$  obteniendo la fórmula para obtener el Path Loss [1].

$$n = \frac{RSSI_0 - RSSI_i}{10 \log_{10} d_i} \quad (F.2)$$

Una vez obtenidos todos los valores necesarios, la fórmula para obtener la distancia se obtiene despejando como se muestra a continuación [1]:

$$d_i = 10^{\frac{RSSI_0 - RSSI_i}{10n}} \quad (F.3)$$

La obtención de la distancia entre las balizas y el dispositivo móvil es esencial para obtener la posición, siendo el primer método empleado en este trabajo para aplicar la Trilateración, explicada más adelante.

### 2.4.2 Distancia a partir del ranging de los Ibeacon

Como novedad introducida a la línea del trabajo anterior, en este trabajo se han incluido balizas en modo Ibeacon, permitiendo, como principal novedad, hallar la distancia entre las balizas y el dispositivo móvil directamente sin necesidad de emplear fórmulas que relacionen ningún otro parámetro [5]. La función Ranging de los IBeacon permite detectar las balizas en una región determinada, junto a la distancia estimada desde el dispositivo móvil a cada módulo BT.

El Ranging devolverá al usuario una matriz de todos los IBeacons encontrados en las inmediaciones del dispositivo junto a sus propiedades (UUID, major, etc.). En base a la proximidad de las balizas se pueden distinguir tres rangos: Inmediato (en un rango de centímetros), Cerca (a un par de metros) y lejos (más allá de 10 metros). El rango máximo de las balizas BLE que empleen el protocolo IBeacon puede alcanzar hasta 450 metros para aquellas que sean de largo alcance, siendo lo habitual unos 70 metros para las más comunes, como se refleja en la figura 6.

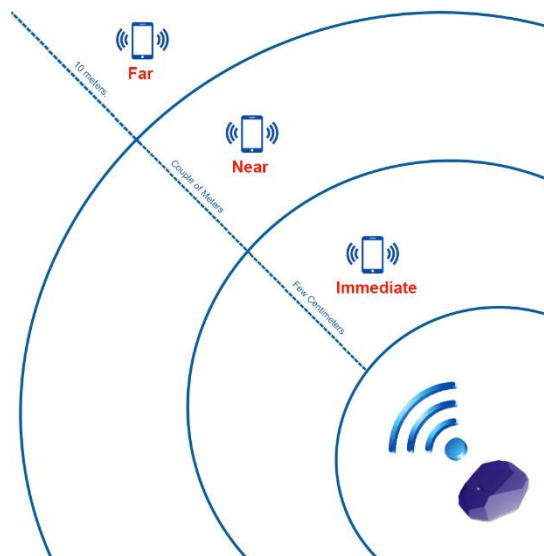
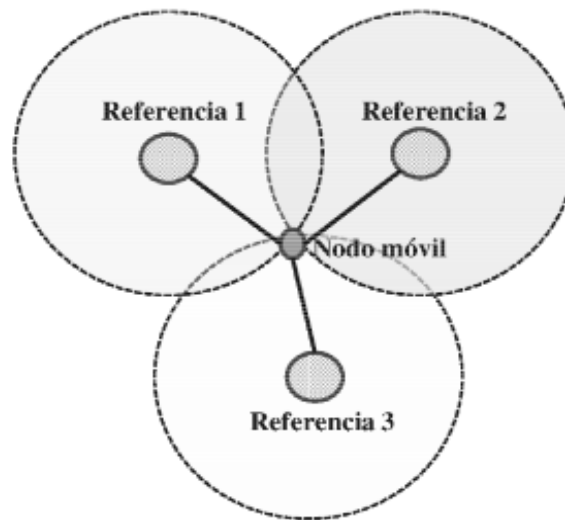


Figura 6: Ranging de los IBeacons

## 2.5 Cálculo de la posición: Trilateración

La trilateración [1] es un método geométrico para determinar la posición de un objeto en base a su distancia a tres puntos conocidos de referencia. A diferencia de la triangulación, en la que además de distancias se utilizan ángulos, en la trilateración se emplean únicamente las distancias.

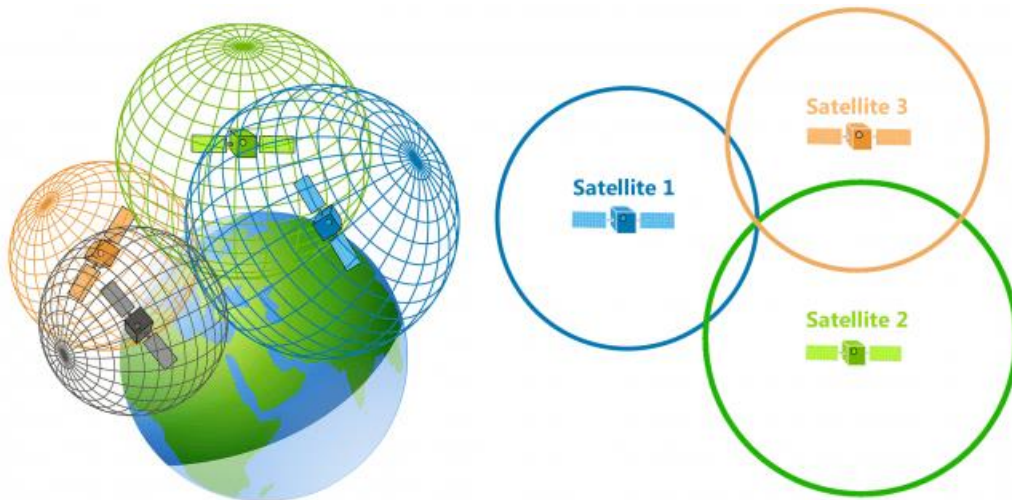


*Figura 7: Representación de la trilateración 2D*

Evidentemente esto es una aproximación sencilla del método aplicable a entornos en 2 dimensiones, utilizando circunferencias. Si extrapolamos la trilateración a un ámbito en 3 dimensiones, es necesario emplear esferas en vez de círculos.

De esta manera, la geolocalización vía GPS permite determinar la posición de un objeto conociendo sus distancias a los satélites que orbitan la Tierra. Como se ha mencionado anteriormente, al tratarse de un entorno 3D se emplean esferas de radio conocido. En base a las distancias conocidas y sus posiciones, se forman 3 esferas que cortan en 3 puntos, 2 en la superficie y una en el espacio. Esto, en consonancia a una cuarta esfera conocida (la Tierra), permite hallar la posición relativa del objeto.





*Figura 8: Representación de la trilateración 3D.*

Este método, sin embargo, es poco eficiente para interiores, ya que se producen alteraciones en las señales que impiden una comunicación efectiva entre los satélites y el dispositivo móvil, desvanecimientos, interferencias, etc. Debido a ello, se tratan de buscar alternativas basadas en este método aplicables a entornos indoor, el propósito principal de este trabajo.

Si nos fijamos en el planteamiento geométrico, si solo conocemos la distancia a un punto, el objeto se podría ubicar en cualquier lugar del círculo formado. Es por ello, que se necesita un segundo punto cuya distancia permita formar otro círculo, pues el objeto ya no se podría ubicar en cualquier lugar del círculo 1 ni del círculo 2. Sólo existen dos puntos que cumplen estas condiciones, las dos intersecciones de los círculos conocidos. El problema aquí radica en que no podemos conocer con los datos que tenemos cuál de los dos puntos es la ubicación verdadera del objeto, y por ello necesitamos un tercer punto.

Conociendo la distancia a un tercer punto, obtenemos un tercer círculo cuya finalidad es interceptar a los otros dos círculos en el único punto que está entre los tres y cumple los criterios de distancia, la posición del objeto.

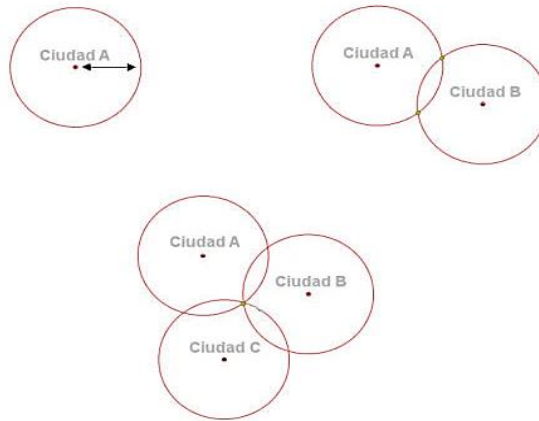


Figura 9: Planteamiento geométrico de la trilateración.

No obstante, esto es un caso particular ideal, pero, lejos de la realidad, suelen ser más comunes situaciones en las que existen únicamente puntos de corte entre dos circunferencias o incluso nunca llegan a cortarse. Los casos posibles [1] se reducen a cuatro:

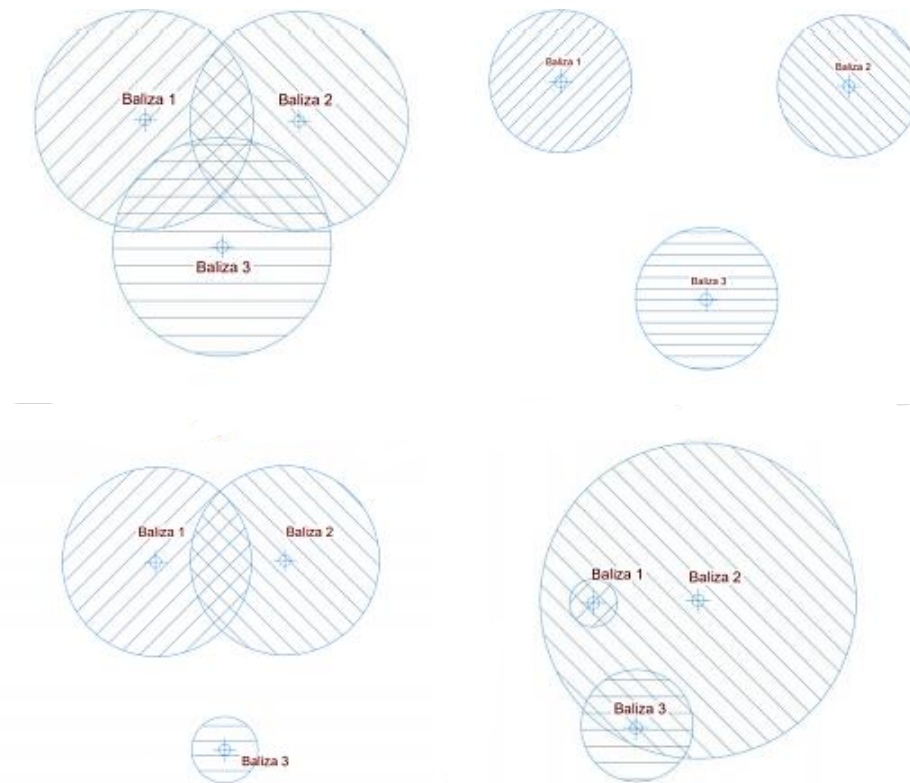


Figura 10: Posibles casos de trilateración.

En el primer caso de la figura 10 se observa el caso ideal descrito previamente: Las tres circunferencias se cortan entre ellas en un punto que resulta ser la ubicación exacta del objeto. No obstante, en el segundo caso podemos ver que ninguna de las circunferencias se llega a cortar debido al escaso tamaño del radio de las circunferencias que forman. En el tercer caso vemos que dos circunferencias se cortan en un punto, pero la tercera no llega a cortarse. Por último, en el último caso encontramos una circunferencia con un radio considerablemente mayor a las anteriores que puede llegar a contener a una o ambas circunferencias formadas por las otras dos balizas.

Si analizamos las circunferencias en parejas, podemos obtener para cada caso 3 puntos, con los cuales se puede formar un triángulo donde se encuentra la posición del objeto. Estos puntos son hallados de la siguiente manera:

- Dos circunferencias se cortan en un punto: Se toma ese punto.
- Dos circunferencias se cortan en dos puntos: Se toma el punto más cercano a la tercera circunferencia.
- Si las circunferencias no se llegan a cortar: Se toma el punto más cercano y equidistante entre ellas.

El resultado aplicable para cada caso lo podemos ver en la figura 11 [1] como solución a los casos planteados en la figura 10:

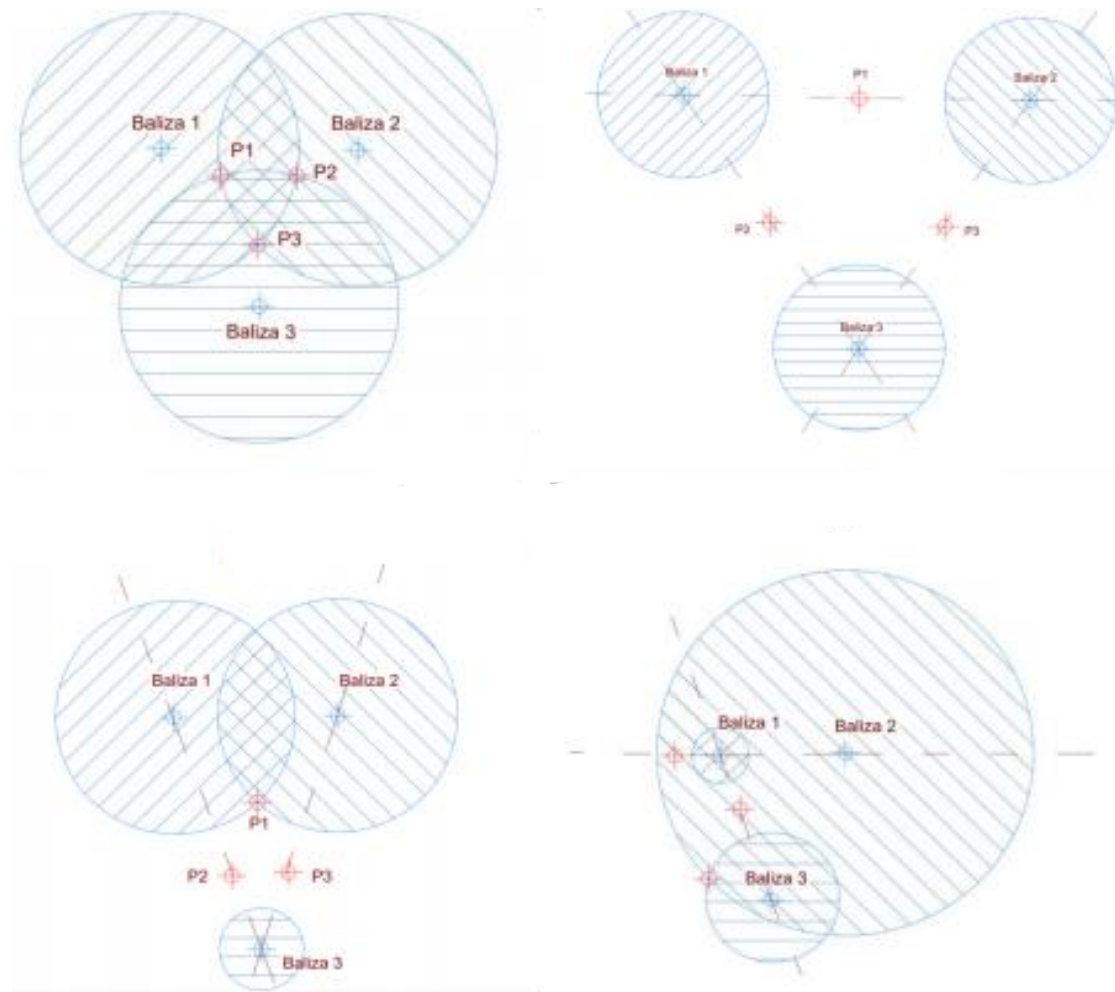


Figura 11: Puntos del triángulo .

El último paso para encontrar la ubicación del objeto sería calcular el centro del triángulo formado por los tres puntos hallados. Esto se puede realizar de cuatro formas distintas, empleando las técnicas de incentro, baricentro, circuncentro y ortocentro para determinar los cuatro posibles centros del triángulo.

Como se describirá en el capítulo (x), realmente el único caso que nos interesa es el primero, ya que es bastante improbable que se produzca cualquiera de las otras tres posibilidades, siempre se dará la situación en la cual se cortan las tres circunferencias en tres puntos.

## 3. ENTORNO DE DESARROLLO

A continuación, se enumerarán y especificarán los principales elementos físicos y de software empleados para llevar a cabo los objetivos prácticos propuestos en este trabajo.

### 3.1 Elementos físicos

En este apartado se describirán los principales elementos que componen el conjunto hardware utilizado en este trabajo para llevar a cabo el proceso físico de generación de la infraestructura y el entorno de trabajo necesario para la obtención de las medidas. Estos elementos hardware se componen básicamente de los módulos BLE empleados para este trabajo, los HM-10, y sus correspondientes baterías LiPo para suministrar energía a los mismos.

#### 3.1.1 DSD TECH HM-10

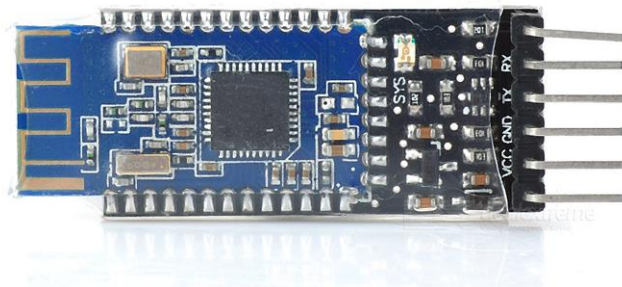


Figura 12: Módulo DSD TECHHM10.

Tomando como referencia [1], en donde se emplearon módulos MLT-BT05, que son dispositivos BLE que incorporan el estándar Bluetooth 4.0. Su principal pega es que por defecto no incluyen la posibilidad de ser programados en el modo iBeacon, para lo cual obliga al programador a cambiar el firmware. El proceso es bastante inestable e inseguro, por lo que la primera decisión a tomar fue buscar alternativas que mantuvieran el mismo hardware, pero con el firmware actualizado y preparado para su programación en modo iBeacon.

Los módulos BLE escogidos reciben la denominación HM-10, respetan la sencillez de configuración a través del IDE de Arduino, así como un bajo coste por dispositivo, de alrededor de 5-6 euros. Está basado en el módulo CC2540, y comparte características en común con él, siendo ambos capaces de funcionar con la versión 4.0 de Bluetooth en modo bajo energía o BLE. Posee 4 pines que conducen a través de la placa base DSD TECH a VCC, GND, TX y RX con un voltaje de trabajo de 3.6 V a 6V.

Cuando se establece la conexión entre el módulo y un dispositivo móvil un LED alojado en la placa que contiene el módulo será el encargado de informar al usuario a través de su encendido intermitente.

La plataforma Arduino IDE se utiliza en este trabajo para programar los módulos HM-10. Esto se realiza a través de los comandos AT, empleados en los módulos Bluetooth, GSM y GPS. Dentro de los comandos AT más comunes empleados para los módulos Bluetooth encontramos:

- **AT:** Comando de prueba para verificar la comunicación bidireccional. El módulo responderá con un OK.
- **AT+STATE?:** El módulo responderá con su estado actual.
- **AT+RESET:** El módulo se reiniciará.
- **AT+VERSION?:** Se obtendrá la versión del firmware del producto.
- **AT+ADDR?:** Sirve para obtener la dirección MAC del módulo.
- **A+PSWD=<N>:** Se establece el pin de 4 dígitos para el emparejamiento Bluetooth, siendo "1234" el pin predeterminado.

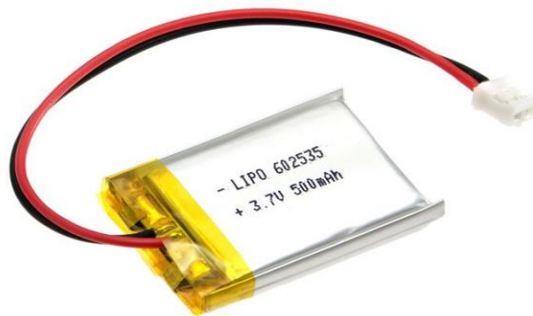
En cuanto a la configuración software [7] del módulo como IBeacon, es necesario seguir una serie de pasos a través de comandos AT:

1. **AT+RENEW:** Reinicio de fábrica del módulo.
2. **AT+RESET:** Se reinicia el HM10.
3. **AT:** Se espera a un OK para confirmar la conexión.
4. **AT+MARJ0X1234:** Se establece como Major number del Ibeacon 0x1234
5. **AT+MINO0XFA01:** Se establece como Minorr number del Ibeacon 0x1234
6. **AT+ADVI5:** Se establece como intervalo de avistamiento a 5 (546 ms)
7. **AT+NAME\_\_:** Se aplica un nombre al módulo HM10.
8. **AT+ADTY3:** Se hace que el módulo sea no conectable para ahorrar batería.
9. **AT+IBEA1:** Activa el modo IBeacon.
10. **AT+DELO2:** Activa el modo broadcast-only de IBeacon para ahorrar batería.
11. **AT+PWRM0:** Activa el modo auto-sleep reduciendo su potencia de 8 a 1.8 mA.
12. **AT+RESET:** Se reinicia finalmente el HM10 para aplicar cambios.

### 3.1.2 Baterías LiPo

Las baterías LiPo (Litio y Polímero) son un tipo de batería recargable compuesta por diferentes celdas en serie, muy habitual en el vuelo de drones, con unas prestaciones superiores a las baterías convencionales.

Existen diferentes modelos en función a la cantidad de energía que pueden almacenar, ofreciendo tasas de descarga elevadas cuando se precisa una demanda elevada de energía, aunque por suerte esto no es necesario en este trabajo, permitiendo funcionar con los módulos activos durante un tiempo cercano a la semana gracias a su ciclo de trabajo basado en el bajo coste energético.



*Figura 13: Batería LiPo de 500mAh/3.7V.*

Cada celda que compone la batería LiPo tiene un voltaje nominal de 3.7 V y permite descargas rápidas contenidas en una batería pequeña y ligera. El principal problema de estas baterías es su escaso ciclo de vida y su facilidad para estallar o prenderse fuego, pudiendo hincharse previamente o perforarse con relativa facilidad.

Las baterías LiPo empleadas en el desarrollo del trabajo son de 3.7 V y 750 mAh para suplir los módulos HM-10, permitiendo su funcionamiento ininterrumpido cerca de una semana. Se conectan directamente a los pines de la base de los BLE y tienen un ciclo de carga relativamente rápido, ideales para el despliegue de las balizas debido también a su ligereza y sinergia con los módulos, formando un tándem compacto y autónomo de dimensiones reducidas.

## 3.2 Elementos Software

En este apartado se describirán los principales elementos que componen el conjunto software utilizado en este trabajo para llevar a cabo el proceso lógico necesario para la obtención de las medidas. Estos elementos se componen básicamente de una herramienta necesaria para la medición en sí a través de un smartphone, TracerPlus, Microsoft Excel para el tratamiento de los datos obtenidos, y MATLAB para la parte de computación y generación de los resultados.

### 3.2.1 Interfaz móvil: TracerPlus Software

Para la obtención de las medidas de los módulos BLE es necesario emplear una aplicación que permita realizar esto, preferiblemente a través de un dispositivo móvil. Tras probar varias apps ya creadas por otros usuarios en Android (*Beacon Scanner* de Nicolas Bridoux, *IBeacons Scanner* de flurp laboratories, etc.) la problemática era siempre la misma: ninguna reunía todas las condiciones necesarias para ser la aplicación de medida escogida para el trabajo. Se precisaba de una aplicación que permitiese leer la información transmitida por los módulos IBeacon, permitiendo obtener, principalmente, su MAC, RSSI y distancia estimada mediante Ranging. Sumado a esto, era preciso que la app permitiese exportar los datos mediante logs a modo de tabla o CSV. Como ninguna reunía todas las condiciones, era necesario crear una app a medida, y esto ha sido parcialmente posible (ya que no es totalmente a medida, pues no permite obtener el comportamiento interno de los módulos en su información) gracias a Tracerplus Software.



Figura 14: TracerPlus Software.



TracerPlus Software [8] es un entorno “sandbox” de desarrollo de aplicaciones para Windows, IOS y Android que, como principal ventaja, permite construir y programar aplicaciones sin la necesidad de conocer ningún lenguaje de programación basándose en una interfaz de usuario bastante atractiva y con un amplio abanico de posibilidades. Las apps se diseñan en la aplicación de escritorio TracerPlus Desktop, y después se pueden desplegar en el dispositivo requerido, escogiendo previamente si se trata de un smartphone Android o IOS, o un equipo con Windows. Para esto, se requiere la descarga adicional de una aplicación, TracerPlus 10, que sirve como contenedor para desplegar el proyecto y utilizar la aplicación. Esta app cuenta con unas aplicaciones de muestra para comprobar de qué es capaz este constructor, como un trackeador de pedidos o un controlador de inventarios.

TracerPlus posee una versión de prueba de 1 mes en la cual se pueden explotar todas las posibilidades de la aplicación; Tras ese mes de prueba, los planes de pago de suscripción son de 39 euros al mes, o 348 euros por dispositivo si se desea una licencia de por vida

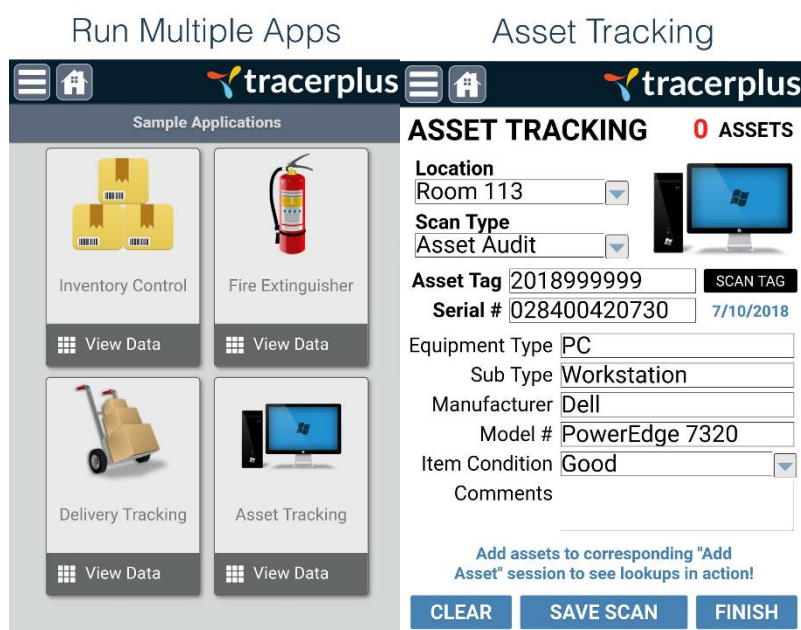


Figura 15: Interfaz de usuario de la app de Android.

La aplicación está pensada para empresas, permitiendo explotar las capacidades del dispositivo móvil, pudiendo construir instrumentos portátiles que empleen tecnologías como códigos de barras, NFC o RFID con fines relacionados con mantenimiento de inventarios, retail, etc. Aunque, como se ha comentado previamente, el límite es la imaginación, las posibilidades son infinitas.

En cuanto a la aplicación diseñada para recibir los beacons, está programada de manera que reciba únicamente IBeacons, permitiendo filtrar entre todas las señales Bluetooth recibidas, lo cual es una gran ventaja de cara a tratar los datos posteriormente, eliminando la necesidad de realizar búsquedas adicionales. En cuanto a las características principales de la aplicación utilizada en este trabajo encontramos a modo de resumen:

- Posibilidad de captar las señales IBeacon en exclusividad, sin necesidad de realizar filtrados adicionales. Esto lo permite realizar la app, permitiendo escoger las señales recogidas en función de si son de tipo Eddystone, IBeacon o Beacons en general.
- La aplicación permite recibir la información esencial de las señales: RSSI, distancia estimada, MAC del módulo, UUID, Major y Minor. Para ello cuenta con un paquete de funciones y características dedicadas a dispositivos Bluetooth. Se podrían haber añadido más características, tales como tiempos y retardos en el viaje y la obtención de las señales, pero no eran necesarias en este trabajo.
- Interfaz visual a modo de tabla para observar las medidas obtenidas en tiempo real. Esto ha sido posible mediante la creación de columnas con los datos correspondientes a la MAC o RSSI, entre otras, formando una tabla conjunta que permite enlazar estos datos obtenidos en tiempo real y mostrarlos de forma simultánea en la pantalla del dispositivo según se actualizan los datos.
- Como se trata de un despliegue, no es necesario descargar ninguna aplicación adicional ni programar su instalación ni paquetes adicionales como drivers. Simplemente es necesario descargar la aplicación creada en el PC mediante un código QR que realiza la descarga del proyecto desde la nube al dispositivo móvil, o bien a través de un cable USB de manera tradicional
- La aplicación posee la capacidad de exportar los datos de cada escaneo como un log en formato txt para tratar los datos posteriormente. Es posible escoger el formato en el que se exportan los datos, desde xlsx hasta CSV separando los datos por punto y coma (;). En este trabajo se ha escogido el formato de texto plano para trabajar los datos de la forma más básica posible y decidir posteriormente si los datos van a ser importados en otra aplicación dedicada al tratamiento de datos (Excel en este caso).
- Finalmente, la app también cuenta con varios toggles y botones creados a antojo del usuario (posición en la app, tamaño, nombre, color...). En este caso se emplean 4 botones:
  - **Escanear:** Comienza el escaneo de forma indefinida hasta que se presione de nuevo el botón para detener el escaneo o bien se pulse el botón “Hecho”.

- **Hecho:** Detiene el escaneo.
- **Limpiar:** Elimina los datos mostrados y almacenados en la tabla para que no se tengan en cuenta a la hora de exportarlos.
- **Submit:** Guarda los datos almacenados en la tabla en el archivo log que posteriormente puede ser exportado.

18:14 4G 76%

**tracerplus**

IbeaconScanner

MAC: 64:69:4E:80:0F:E0

Datos: 0201061AFF4C00021574278BDAB64

UUID: 74278bda-b644-4520-8f0c-720eaf059

Major: 4096

Minor: 7

RSSI: -58

Distancia: 0,84

MAC	Datos	UUID	Major	Minor	RSSI	Distancia
64:69:4E:80:0F:E0	0201061AFF4C00021574278BDAB64	74278bda-b644-4520-8f0c-720eaf059	4096	7	-58	0,59
64:69:4E:80:0F:E0	0201061AFF4C00021574278BDAB64	74278bda-b644-4520-8f0c-720eaf059	4096	7	-56	0,84
64:69:4E:80:0F:E0	0201061AFF4C00021574278BDAB64	74278bda-b644-4520-8f0c-720eaf059	4096	7	-58	0,59
64:69:4E:80:0F:E0	0201061AFF4C00021574278BDAB64	74278bda-b644-4520-8f0c-720eaf059	4096	7	-56	0,34
64:69:4E:80:0F:E0	0201061AFF4C00021574278BDAB64	74278bda-b644-4520-8f0c-720eaf059	4096	7	-53	0,23

Hecho Submit Limpiar Escanear

Figura 16: Interfaz del IBeacon Scanner.

### 3.2.3 Organización de los datos: Microsoft Excel

Para este trabajo, los datos exportados por la aplicación móvil han sido agrupados en un archivo común que contiene todas las medidas captadas durante la fase de medición. Esto se ha hecho por comodidad y para evitar trabajar con una gran cantidad de ficheros al unificarlos todos para su posterior análisis y uso en el algoritmo de trilateración.

La manera más sencilla de realizar esto es importar todos los ficheros empleando una aplicación dedicada principalmente para este tipo de tareas, un gestor de hojas de cálculo, por ejemplo. De este modo, se evita dar complejidad al trabajo mediante el uso de herramientas o toggles dedicadas al trabajo con tablas y ficheros, evitando escribir

bloques de código para este cometido. Como se acaba de comentar, esto se podría haber realizado también empleando varios IDEs con diferentes lenguajes de programación (Java, Python, MATLAB, etc.), pero en su lugar se ha optado por el uso de Microsoft Excel para esta parte del trabajo.

Se ha decidido emplear este programa para el tratado de los datos debido a las posibilidades que ofrece, permitiendo filtrar datos exageradamente erróneos donde se obtenían medidas absurdas y, como principal motivo, para ordenar y agrupar los datos según se requiera y obtener errores medios en las medidas e histogramas de estos sin necesidad de escribir código para llevar a cabo esta tarea.



Figura 17: Interfaz de Microsoft Excel.

### 3.2.4 MATLAB

Matlab es un entorno de desarrollo integrado o IDE con un lenguaje de programación propio (lenguaje M) empleado para el cómputo numérico. Su nombre proviene de la abreviatura de MATrix LABoratory (laboratorio de matrices) y fue lanzada inicialmente en 1984 por MathWorks. Está disponible para las plataformas Windows, MAC OS, Linux y Unix mediante el pago de licencias de distribución.

Como principal atractivo, MATLAB permite entre sus prestaciones el diseño de scripts para:

- Cálculo computacional.
- Implementación y desarrollo de algoritmos.
- Análisis y visualización de los datos.
- Simulación y prueba de prototipos.
- Desarrollo de aplicaciones que requieran una GUI o interfaz gráfica de usuario.

MATLAB se emplea en el desarrollo de este trabajo para tratar los datos una vez ordenados con Microsoft Excel. Esto se puede realizar fácilmente debido a una poderosa característica que posee MATLAB, la posibilidad de importar hojas de cálculo.

La inclusión de MATLAB en este proyecto permite realizar los cálculos numéricos necesarios en la implementación del algoritmo de trilateración, así como realizar el diseño en 2D del plano del entorno de medición y los puntos obtenidos en cada medida, pudiendo representarlos visualmente para analizar los resultados posteriormente con relativa facilidad.

## 4. ALGORITMO DE LOCALIZACIÓN: Propuesta inicial y Nueva propuesta

En este capítulo se introduce el algoritmo de localización en interiores propuesto en el trabajo de fin de grado anterior [1] sirviendo como preámbulo del nuevo modelo implementado en el trabajo de fin de grado actual. A modo de resumen, el algoritmo empleado en el anterior trabajo permite posicionar con cierta precisión un objeto en un área formado por varias balizas siempre y cuando el dispositivo se encuentre dentro de dicha área, elevando el error cometido considerablemente cuando sucede el caso contrario.

Posteriormente, se expondrá el funcionamiento del nuevo algoritmo empleado en este trabajo, describiendo sus características, metodología y novedades incluidas para reforzar el trabajo anterior.

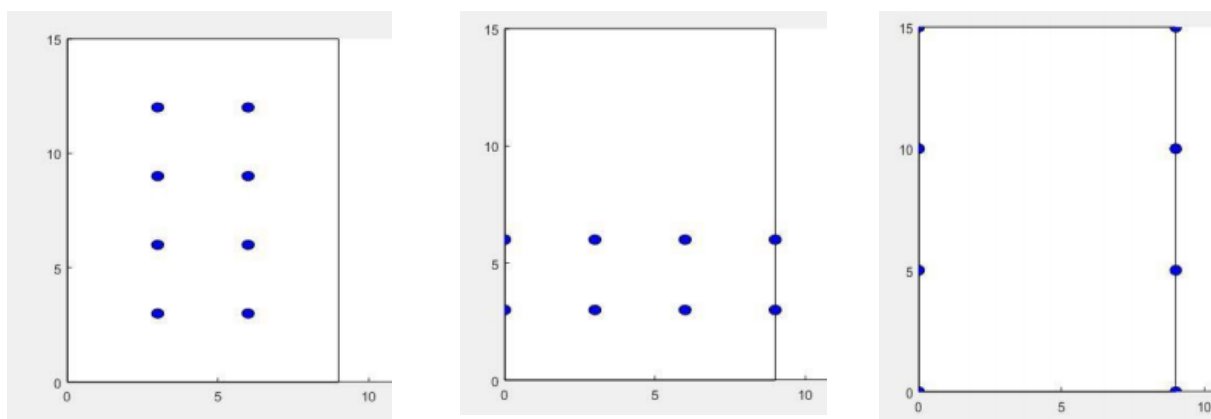
Como se mencionó en el primer capítulo, el propósito de este trabajo es introducir una nueva tecnología para comprobar si existe mejora en los resultados de posicionamiento en interiores por medio de trilateración, pudiendo añadir un nuevo método de obtención de la distancia estimada de forma directa mediante el ranging de los IBeacons y comparando resultados.

Además, para darle más riqueza al estudio, se han realizado medidas con una determinada orientación (Norte, Sur, Este y Oeste) para comprobar si, como mencionaba Carlos en su TFG a modo de conclusión, el cuerpo actúa como pantalla y esto influye directamente en los resultados obtenidos en la medida.

### 4.1 Escenario empleado en ambas propuestas

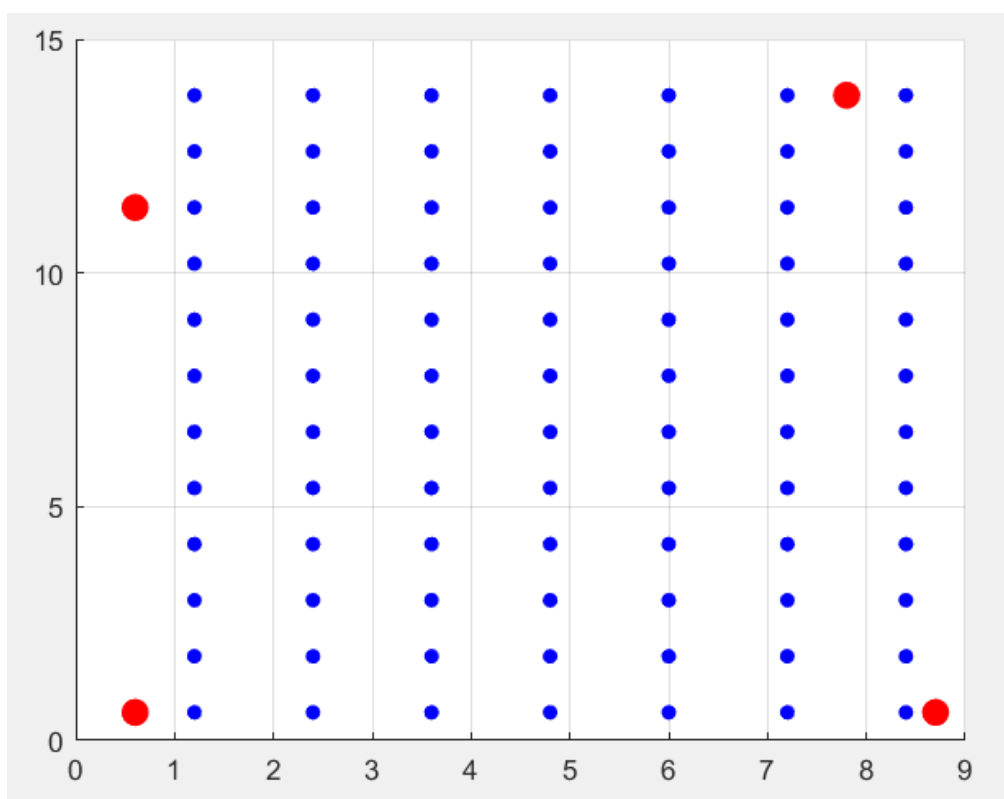
En el trabajo anterior propuesto por Carlos se hace uso de 8 balizas, colocadas en la misma aula donde se realiza este mismo proyecto, la cual mide 15 metros de largo y 9 metros de ancho. La distribución de las balizas se realizaron siguiendo tres distribuciones distintas, planteando tres escenarios jugando con la posición de los módulos Bluetooth, mostrados en la figura 18.

En el trabajo actual se emplean 4 balizas, colocadas de forma bidimensional intentando formar un rectángulo. Lamentablemente, como se podrá comprobar en la figura 19, la disposición de las balizas no forma un rectángulo exactamente, sino un polígono. Esto se debe principalmente a que las balizas necesitan estar apoyadas en un punto alto, encima de una estantería, por ejemplo, y estas estaban colocadas en el aula de una forma que no resulta ser un rectángulo.



*Figura 18: Escenarios del trabajo anterior*

En el primer escenario planteado en el TFG anterior se dispusieron las balizas manteniendo una separación de 3 metros entre ellas. En el segundo se hizo lo mismo pero disponiendo parte de ellas en las paredes, para comprobar la influencia de la difracción o reflexión. Finalmente, en el último escenario colocaron las balizas manteniendo una separación lo más alejadas posibles entre ellas.



*Figura 19: Escenario del trabajo actual*

En el trabajo anterior se tomaron medidas en puntos aleatorios durante 10 segundos estando inmóvil para el caso estático. En el primer escenario se tomaron 9, en el segundo 5 y en el tercero 4.

## 4.2 Resultados de la propuesta anterior

Tras la obtención de los errores en las medidas, el trabajo anterior concluye con la afirmación de que los resultados mejoran y los errores se hacen menores en los casos donde las medidas se realizaron dentro del rectángulo delimitado por las balizas. Esta premisa se confirma en los tres escenarios, aportando al trabajo el hecho de que la compensación de las reflexiones/refracciones de las señales se puede realizar mediante la colocación de las balizas pegadas a las paredes, obteniendo mejores resultados situando la medida pegada a una de las paredes en el segundo escenario.

Por último, la toma de medidas del tercer escenario aporta al estudio una posible deducción en la disposición de las balizas, de tal manera que un móvil tenga al menos 3 balizas en un rango entre 4 y 6 metros para que los resultados se aproximen a la ubicación real del dispositivo.

Para darle riqueza al estudio, se añadió un último estudio del algoritmo de posicionamiento mediante el empleo de los mismos escenarios, pero permaneciendo el objeto en constante movimiento, trazando un camino conocido previamente. El resultado cumple con las conclusiones anteriores, ya que el trayecto estimado resulta próximo al trayecto real cuando las medidas se realizan en el interior del rectángulo formado por la disposición de las balizas, alejándose de la realidad cuando se toman fuera del rectángulo descrito

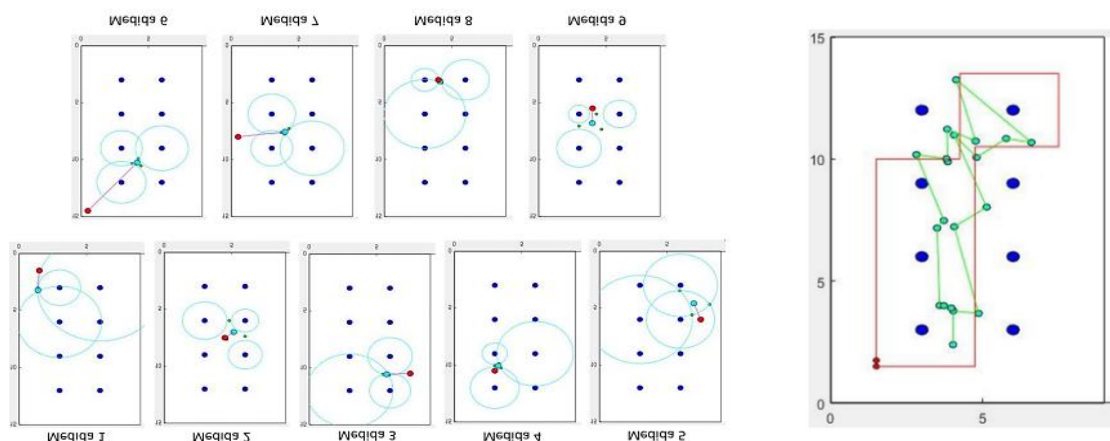


Figura 20: Resultados del primer escenario.



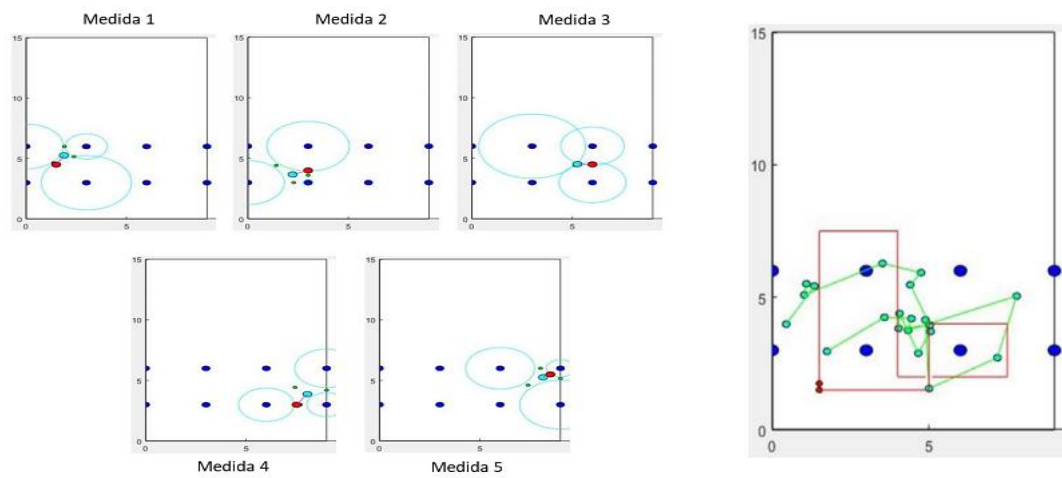


Figura 21: Resultados del segundo escenario

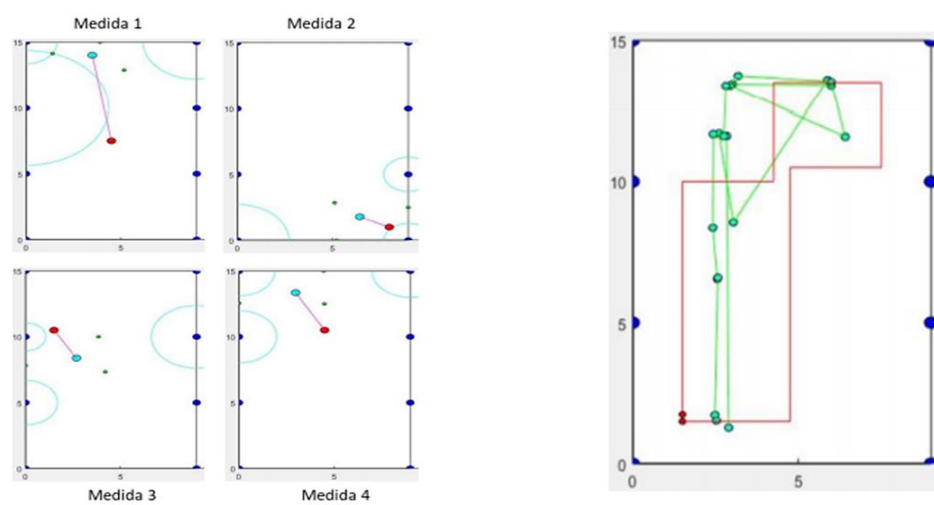


Figura 22: Resultados del tercer escenario

## 4.3 Nueva propuesta del algoritmo de localización en interiores

Como se ha comentado en el apartado anterior, el escenario empleado en la nueva propuesta para el algoritmo de posicionamiento indoor es el mismo. Partiendo de la misma aula, este trabajo se distingue del anterior en la inclusión de la tecnología IBeacon en las balizas BT y en que sólo se realiza una única disposición de las balizas a lo largo de todo el estudio, pero se incrementan considerablemente el número de puntos de medida, formando una malla de 84 puntos distribuidos de forma equidistante a lo largo del aula realizando 8 mediciones de 10 segundos cada una: 2 con orientación norte, 2 con orientación sur, 2 con orientación este y 2 con orientación oeste. En total, 672 medidas.

### 4.3.1 Tratamiento de los datos y errores en la medida

Previo a la implementación del algoritmo de posicionamiento, en primer lugar, hace falta realizar un tratamiento de los datos y obtener los mismos en un fichero conjunto de forma ordenada, que es lo que se va a explicar a continuación. Los ficheros obtenidos con la aplicación en cada medida constan de 3 columnas seleccionadas de las 5 posibles: MAC, RSSI y Distancia. El primer paso en el tratamiento de los datos ha sido agrupar todas las medidas en un gran fichero conjunto que contuviera todos los datos. Esto se ha realizado, como se ha comentado previamente, mediante Microsoft Excel.

Hay que tener en cuenta que cada fichero ha sido renombrado a propósito con la coordenada del punto de medida y su orientación, con la siguiente forma: “1\_1\_N1”, que indica que es la primera medida con orientación norte en el punto 1.1.

En el proceso de ir importando los datos de cada fichero se han añadido nuevas columnas a las tres ya existentes para poder identificar mejor los datos de cada medida. Estas nuevas columnas son:

- **Orientación:** Indica con una letra la orientación de la medida (N,S,E,W). Esto se añadió para dar riqueza al estudio y poder sacar conclusiones de cara a posibles estudios posteriores.
- **Nº de medida:** Añadido para posteriormente agrupar los datos y reorganizarlos.
- **Xmedida e Ymedida:** Indica el punto en el mapa donde se ha realizado la medida. Este punto es orientativo, no incluye la medida en metros, y va de 1 a 7 para el eje X y de 1 a 12 para el eje Y (por ejemplo, 1.3).

- **XrMedida e YrMedida:** El punto anterior es orientativo, y sirve para conocer rápidamente en qué punto de medida nos encontramos, pero para el algoritmo es necesario conocer la distancia real de cada baliza al punto de medida, por lo que se añaden estas dos columnas que transforman el punto anterior en su medida real en metros para cada coordenada X e Y del punto de medida.
- **XBeac e YBeac:** Indica la posición en metros del beacon del que se ha obtenido la medida (esto se sabe en función de su MAC conocida).

Una vez agrupados todos los datos, se observó en los resultados preliminares de la toma de medidas ciertos comportamientos extraños en las mismas a través de diversos cálculos en los errores de las distancias, apoyados en diferentes gráficos e histogramas, por lo que se decidió realizar un filtrado de los datos. Como principal consecuencia, debido a errores en la medida (ya sea por interferencias o la propia aplicación móvil), se obtienen RSSI inferiores a -100dbm o distancias superiores a 30 metros (el doble del aula), por lo que esos datos son incoherentes y se da por hecho que son totalmente incorrectos.

Tras realizar el filtrado, eliminando las medidas con las RSSI o distancias erróneas comentadas previamente, posteriormente se hallan las distancias obtenidas y su correspondiente error. Esto se podría haber realizado con MATLAB pero, para dotar de simplicidad al trabajo ya que se está empleando una potente herramienta como Excel, se ha realizado aquí. En este proceso se distinguen dos partes principales:

- La distancia teórica, calculada mediante la fórmula del path loss (x) comentada previamente.
- La distancia obtenida directamente de los beacons.

Ambos valores se comparan con un valor añadido, “DistReal”, que es la distancia real desde el punto de medida hasta el IBeacon del que se obtiene la medida. Este valor se ha calculado mediante la fórmula de distancia entre dos puntos (x1, y1) y (x2, y2).

En la siguiente figura se muestra un ejemplo del fichero resultante usando como ejemplo un número de medidas.

MAC	RSSI	Distancia	Orientación	NºMedida	Xmedida	Ymedida	XrMedida	YrMedida	Xbeac	Ybeac	DistReal	ErrorReal	DistTeórica	ErrorTeórico
FC:45:C3:23:FC:29	-85	7,85	E	1	1	1	1,2	0,6	7,8	13,8	14,7580487	-6,91	2,8283635	-11,929685
FC:45:C3:23:FC:29	-85	7,12	E	1	1	1	1,2	0,6	7,8	13,8	14,7580487	-7,64	2,8283635	-11,929685
FC:45:C3:23:FC:29	-85	6,45	E	1	1	1	1,2	0,6	7,8	13,8	14,7580487	-8,31	2,8283635	-11,929685
FC:45:C3:23:FC:29	-86	5,83	E	1	1	1	1,2	0,6	7,8	13,8	14,7580487	-8,93	3,0313604	-11,726688
FC:45:C3:23:FC:29	-86	5,83	E	1	1	1	1,2	0,6	7,8	13,8	14,7580487	-8,93	3,0313604	-11,726688
FC:45:C3:23:FC:29	-86	5,27	E	1	1	1	1,2	0,6	7,8	13,8	14,7580487	-9,49	3,0313604	-11,726688
FC:45:C3:23:FC:29	-86	4,76	E	1	1	1	1,2	0,6	7,8	13,8	14,7580487	-10,00	3,0313604	-11,726688
FC:45:C3:23:FC:29	-90	11,49	E	2	1	1	1,2	0,6	7,8	13,8	14,7580487	-3,27	3,99988	-10,758169
FC:45:C3:23:FC:29	-90	10,47	E	2	1	1	1,2	0,6	7,8	13,8	14,7580487	-4,29	3,99988	-10,758169
FC:45:C3:23:FC:29	-89	9,52	E	2	1	1	1,2	0,6	7,8	13,8	14,7580487	-5,24	3,7320256	-11,026023
FC:45:C3:23:FC:29	-88	9,52	E	2	1	1	1,2	0,6	7,8	13,8	14,7580487	-5,24	3,4821082	-11,27594
FC:45:C3:23:FC:29	-85	9,52	N	3	1	1	1,2	0,6	7,8	13,8	14,7580487	-5,24	2,8283635	-11,929685
FC:45:C3:23:FC:29	-85	6,45	N	3	1	1	1,2	0,6	7,8	13,8	14,7580487	-8,31	2,8283635	-11,929685
FC:45:C3:23:FC:29	-85	6,45	N	3	1	1	1,2	0,6	7,8	13,8	14,7580487	-8,31	2,8283635	-11,929685
FC:45:C3:23:FC:29	-84	5,27	N	3	1	1	1,2	0,6	7,8	13,8	14,7580487	-9,49	2,6389604	-12,119088

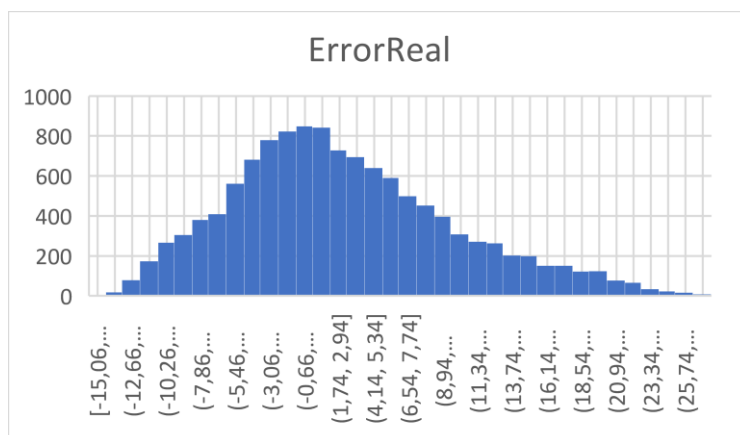
Figura 23: Fichero resultante de agrupar todas las medidas.

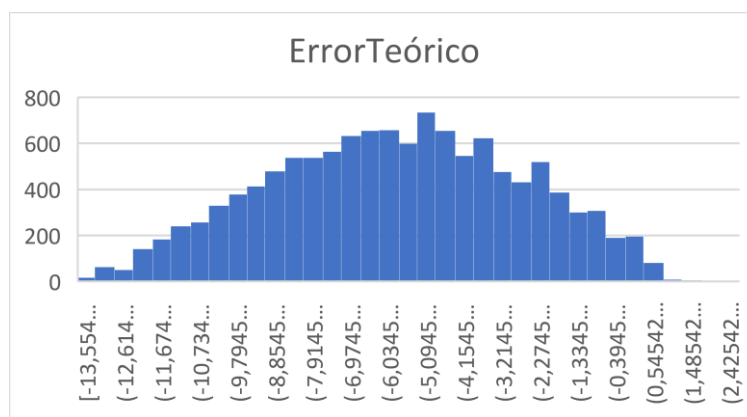
Gracias a esta tabla resultante con los datos de todas las medidas, se pueden obtener los errores promedio en la obtención de la distancia mediante ambos métodos, así como el error promedio para cada orientación y un histograma de los mismos.

Método Empleado	Error medio obtenido
Path Loss	2,66 metros
Ranging	-5,63822 metros

Tabla 5: Errores promedio.

Como se puede observar en la tabla anterior, el error promedio de todas las medidas tomadas a lo largo del estudio es inferior si se opta por el segundo método, empleando directamente la distancia estimada por los IBeacons. Esta diferencia es bastante favorable para el estudio, pudiendo comprobar que existen 3 metros de diferencia a favor de esta tecnología. A continuación, se muestra un histograma con ambos errores representados mediante una campana de Gauss.





Figuras 24 y 25: Histogramas de los errores promedio.

En cuanto a la influencia de la orientación del dispositivo de medida en la toma de estas, se recogen los errores en la siguiente tabla, observando que no existe prácticamente variación entre una orientación u otra, descartando la posible influencia del posicionamiento descrito como motivo de conclusión en el TFG anterior.

Método Empleado	N	S	E	W
Path Loss	-5,571 m	-5,666 m	-5,59 m	-5,72 m
Ranging	2,74 m	2,64 m	3,03 m	2,25 m

Tabla 6: Errores promedio en función de la orientación.

El último paso en el tratamiento de los datos consiste en la agrupación de las medidas y su reordenación para facilitar la lectura de las mismas con MATLAB en la implementación del algoritmo de trilateración. La finalidad de este proceso consiste en la obtención de una tabla final que contenga las medidas ordenadas de la siguiente manera

```
tabla=xlsread("probando.xlsx");  
tam = size(tabla);  
tam=tam(1);  
  
%Variables auxiliares y contadores  
medida_aux=1;  
RSSI=0;  
dist=0;  
cont=0;
```

En primer lugar, se lee el fichero resultante anterior y se obtiene su tamaño total con el comando `size` para poder iterar sobre los valores a continuación. Para los promedios se han de definir dos variables inicializadas a 0 que sirvan para ir acumulando el sumatorio de distancias y RSSIs.

```
%Ordeno los datos con su sumatorio por cada medida en cada punto  
for i=1:tam  
    MAC=tabla(i,1);  
    medida=tabla(i,5);  
    x=tabla(i,6);  
    y=tabla(i,7);  
    if medida==medida_aux  
        RSSI=RSSI+tabla(i,2);  
        dist=dist+tabla(i,3);  
        cont=cont+1;  
        x_aux=tabla(i,6);  
        y_aux=tabla(i,7);  
    else %Cuando llega a una nueva medida  
        j=j+1;  
        RSSI=RSSI/cont;  
        dist=dist/cont;  
        datos(j,1)=MAC;  
        datos(j,2)=RSSI;  
        datos(j,3)=dist;  
        datos(j,4)=x_aux;  
        datos(j,5)=y_aux;  
        RSSI=0+tabla(i,2);  
        dist=0+tabla(i,3);  
        cont=1;  
        medida_aux=medida;  
    end  
end
```

Mediante este bloque de código, obtenemos una nueva tabla llamada (datos) que contiene el promedio de distancia y RSSI de cada medida, pero sin ordenar como queremos en grupos de 4. Para ello, se itera sobre la tabla inicial y se guardan los valores de MAC, nº de medida y coordenadas para saber en qué punto y número de medida de cada MAC estamos. Como se ha mencionado previamente, se compara el valor de la medida actual con un variable auxiliar de medida y, si coincide, estamos en el mismo nº de medida (y por tanto punto de medida) y se suma el valor de la RSSI y distancia de la medida al sumatorio inicializado a 0, aumentando también el contador para realizar el promedio.

Una vez se llegue a un número de medida que no coincida con el número anterior, significa que se ha cambiado de medida y, por tanto, se ha de realizar el promedio de RSSI y distancia dividiendo el sumatorio entre el contador. El valor de la MAC, los promedios y las coordenadas X e Y se insertan en una nueva tabla que contiene los datos desordenados.

Finalmente, se inicializan los sumatorios de la RSSI y distancia con los valores actuales, se inicia el contador a 1, y se copia el valor de la medida actual en la variable auxiliar. Y se repite el proceso de nuevo con la medida actual.

Con esto, obtenemos una tabla con los datos promediados para cada MAC en cada medida y en cada punto, obteniendo 8 puntos para cada MAC en cada punto de medida. La única pega es que estos datos no están ordenados, así que finalmente sólo queda ordenarlos en grupos de 4

```
%Ahora ordeno las medidas por MAC y punto
tamdatos = size(datos);
tamdatos=tamdatos(1);
%Variables auxiliares y contadores
medida_aux=1;
RSSI=0;
dist=0;
cont=0;
MACS=[1 2 4 5];
cont_MAC=1;
x_aux=1;
y_aux=1;
%Nueva tabla
final=zeros(2000,5);
j=0;
exito=0;
```

Como paso inicial, se lee la tabla obtenida en el paso anterior y se obtiene su tamaño para iterar sobre los datos. También se definen valores inicializados a 0 y un contador para las MAC leídas, así como un vector que contiene el número de la MAC. Los datos finales reordenados se insertarán en una tabla “final” que contendrá dichos datos.

```
for l=1:700
    exito=0;
for i=1:tamdatos
    MAC_aux=MACS(cont_MAC);
    MAC=datos(i,1);
    RSSI=datos(i,2);
    dist=datos(i,3);
    x=datos(i,4);
    y=datos(i,5);
    if MAC==MAC_aux && x==x_aux && y==y_aux
        j=j+1;
        datos(i,1)=0; %Pongo la MAC de la tabla que leo a 0 para la siguiente pasada
        final(j,1)=MAC;
        final(j,2)=RSSI;
        final(j,3)=dist;
        final(j,4)=x;
        final(j,5)=y;
        cont_MAC=cont_MAC+1;
        exito=exito+1;
    if (cont_MAC>length(MACS)) %Cuando la ultima MAC se ha leído
        cont_MAC=1;
    end
end
end
end
```

En primer lugar, se establece una MAC auxiliar para ir “recolectando” cada MAC en cada iteración y guardar sus datos en las variables ya comentadas. En cada iteración se comprueba mediante un if si la MAC actual coincide con la auxiliar y si los puntos de medida son los mismos (ya que podría ser la misma MAC pero de otro punto). En ese caso, se pone esa MAC a 0 para no tener en cuenta ese valor en la próxima pasada y se copian en la tabla final los valores de la medida.

Ahora viene lo importante, para tener ordenados los datos en grupos de 4, necesitamos que el siguiente valor buscado sea el de la siguiente MAC, es decir, si estábamos en la MAC 1, ahora buscamos la número 2. Para ello, aumentamos el contador de MACs que itera sobre el vector que contiene dichos números y se suma 1 al contador de éxitos, que lleva la cuenta del número de MACS llevamos para cada grupo (hasta un máximo de 4). Al aumentar el contador que itera sobre las MACs, hay que tener en cuenta que buscamos grupos de 4, entonces si el contador +1 es mayor que la longitud del vector, hay que inicializarlo a su valor inicial, la MAC 1. Esto significa que, si hemos leído la MAC 4 exitosamente, y ahora el contador tiene valor 5, siendo el valor del vector 4 MACS, significa que se reinicia la cuenta ya que hemos conseguido un grupo de 4 completo.



```

        if exito~=4
            j=j-exito;
            y_aux=y_aux+1;
            cont_MAC=1;
            if y_aux>12
                x_aux=x_aux+1;
                y_aux=1;
            end
        end
    end
end

%Finalmente guardo en un archivo .txt
writematrix(final,'trilaterar.xlsx')

```

Como último paso, queda comprobar si hemos cambiado de punto, ya que el proceso anterior se realiza en cada punto. Si se ha cambiado de punto, se inicializa el contador de MACS a 1 para buscar el primer valor del grupo de 4 y se suma 1 al valor de la Y auxiliar. Si Y auxiliar llega a 12, significa que hemos recorrido la primera columna de medidas (de la 1.1 a la 1.12), entonces la Y auxiliar se inicializa a 1 y la X auxiliar aumenta su valor en 1 (pasamos al punto inicial de la columna siguiente, en este caso de ejemplo sería el 2.1). Tras este proceso, se guarda dicha tabla en una hoja de datos Excel para pasar finalmente al algoritmo de trilateración.

El resultado final de esta etapa de reordenación de los datos se muestra a continuación en la siguiente figura:

MAC	RSSI	Dist	X	Y
1	-85,5714286	6,158571429	1	1
2	-72	12,88857143	1	1
4	-87	9,004	1	1
5	-76	18,302	1	1
1	-89,25	10,25	1	1
2	-70,6666667	18,17833333	1	1
4	-82,5	3,73	1	1
5	-80,625	11,66375	1	1
1	-84,6	6,592	1	1
2	-70,25	14,475	1	1
4	-74,6666667	8,521666667	1	1
5	-79,6666667	5,663333333	1	1
1	-88,2	7,12	1	1
2	-70	17,1775	1	1
4	-79,625	8,63125	1	1
5	-77	9,54	1	1

Figura 26: Datos reordenados para la trilateración.

### 4.3.2 Algoritmo de posicionamiento

Partiendo de un escenario predefinido y conocido, se realiza un despliegue de X balizas colocadas a lo largo de la estancia de una forma determinada, las cuales se pueden representar sobre una matriz de dimensión  $m \times n$ . De este modo, se puede definir una matriz MAC con los identificadores de cada baliza  $X_n$  donde cada elemento MAC de la matriz representa la dirección física de cada baliza en el entorno definido [1].

Conforme con lo anterior, se puede hacer lo mismo para definir una nueva matriz RSSI en la que se almacena el valor medio de la potencia recibida en la posición definida de acuerdo con el escenario planteado y la matriz anterior [1].

Como novedad en este trabajo, se ha introducido una nueva característica inherente a la tecnología IBeacon, el ranging. Este parámetro nos permite obtener la distancia estimada al módulo sin necesidad de realizar ningún cálculo adicional. De esta forma, se puede definir una nueva matriz DIST en la que se almacena el valor medio de la distancia estimada de acuerdo con la posición del módulo en el escenario planteado.

$$DIST_{(x,y)} = \begin{pmatrix} dist11 & \dots & dist1n \\ \vdots & \ddots & \vdots \\ distm1 & \dots & distmn \end{pmatrix} = (DIST)_{m \times n} \quad (F.4)$$

Las matrices resultantes se utilizan para el método de trilateración, obteniendo tres puntos de referencia, P1, P2 y P3. Estos tres puntos se obtienen siguiendo un patrón común para ambos métodos, la distancia en función de la potencia. Dado que la potencia recibida por el dispositivo es directamente dependiente de la distancia, para escoger los tres puntos para realizar la trilateración es necesario quedarse con los tres con mayor potencia recibida RSSI. En el caso del estudio del ranging, es necesario quedarse con los tres con menor distancia estimada.

Este método se implementa realizando una triple iteración sobre las matrices de RSSI y Distancia iniciales, quedándose en cada iteración con el módulo Bluetooth Px con mayor RSSI o menor distancia estimada.

Dado que la distribución de las balizas en el escenario no es la misma en los trabajos, se ha optado por emplear otro código distinto al empleado en [1], utilizando ciertos procedimientos aplicables a este trabajo, que ya fueron empleados en el trabajo anterior. Por ejemplo, ya que no se emplean más que 4 balizas en este trabajo, no es necesario tener en cuenta supuestos casos de balizas halladas en línea recta como se realizó en el TFG anterior, dado que en este caso es imposible que las 3 balizas halladas se encuentren en una línea recta.

Una vez obtenidos los 3 puntos, se halla mediante el método path loss comentado anteriormente la distancia estimada al dispositivo móvil, rx. En el caso del segundo estudio, este paso es omitido dado que ya conocemos previamente dicha distancia a

través del Ranging de los IBeacons. Estas tres distancias  $r_1$ ,  $r_2$  y  $r_3$  corresponden a los radios de las circunferencias asociadas a cada punto.

En el trabajo de Carlos se realizaban varios estudios hallando la posición del dispositivo móvil a través de los diferentes centros del triángulo formado por los puntos de corte de las circunferencias, es decir, desde un punto de vista geométrico. En este proyecto se ha optado por un método analítico dada la existencia de únicamente 4 módulos BLE, pues se va a verificar siempre el primer caso de los 4 comentados en apartados anteriores: Siempre se cortan las circunferencias.

Es por esto por lo que el código no es el mismo empleado en [1], pues siempre nos situaremos en el primer caso, entonces no es necesario hallar todos los centros del triángulo ni hallar puntos adicionales entre ambas circunferencias, ya que ese proceso está más simplificado en este escenario de estudio, y es posible prescindir de gran parte del código anterior.

Finalmente se procede a la trilateración partiendo de los siguientes datos obtenidos:

- Tres posiciones conocidas  $(x_1, y_1)$ ,  $(x_2, y_2)$  y  $(x_3, y_3)$
- Tres puntos de distancia  $r_1$ ,  $r_2$  y  $r_3$ .

Según el teorema de Pitágoras, el cálculo de la posición del punto de intersección es el punto desconocido  $(x_0, y_0)$ :

$$\begin{aligned}(x_1 - x_0)^2 + (y_1 - y_0)^2 &= r_1^2 \\(x_2 - x_0)^2 + (y_2 - y_0)^2 &= r_2^2 \\(x_3 - x_0)^2 + (y_3 - y_0)^2 &= r_3^2\end{aligned}\tag{F.5}$$

La obtención del punto de localización del dispositivo móvil coincide con el punto desconocido de la intersección.

En el siguiente apartado se realizará una descripción del método comentado previamente desarrollado de forma particularizada para el entorno de cómputo utilizado, MATLAB.

### 4.3.3 Implementación

Lo primero que hace el algoritmo implementado en MATLAB es leer los datos correspondientes a las medidas contenidas dentro de la tabla obtenida en el proceso de tratado de los datos., como se hizo en dicho apartado. Como se precisan grupos de 4, se leen los datos de los otros tres módulos siguientes aumentando el contador de la iteración una, dos y tres posiciones.

Dichos datos de potencia se almacenan en una matriz P y las MACS en una matriz MACS para el método Path loss, y en el segundo estudio se realiza lo mismo, pero en vez de una matriz P, se crea la matriz DIST, con los valores de distancia de las 4 balizas.

```
|while i<=tamfich
    %Obtengo las 4 distancias y sus MACS para identificarlas
    r1=medidas(i,3);
    MAC1=medidas(i,1);
    r2=medidas(i+1,3);
    MAC2=medidas(i+1,1); %Sumo posiciones para coger los grupos de 4 en 4
    r3=medidas(i+2,3);
    MAC3=medidas(i+2,1);
    r4=medidas(i+3,3);
    MAC4=medidas(i+3,1);

    R=[r1 r2 r3 r4];           %Organizo las Distancias y MACS en vectores
    MACS=[MAC1 MAC2 MAC3 MAC4];
```

Es necesario también conocer el punto en el que se está realizando la medida, por lo que hace falta almacenar en dos variables las coordenadas del punto de medida, que corresponden a la cuarta y quinta columna, y pasarlo a la escala real, en metros. Esto es común a ambos métodos.

```
xpos=medidas(i,4);
xr=1.2*xpos;
ypos=medidas(i,5);
yr=1.2*ypos-0.6;
```

A continuación, se procede a hallar los tres puntos necesarios para realizar la trilateración. En el primer método, como se ha comentado en el apartado anterior, se hallan los tres puntos en función de su RSSI, escogiendo los tres con mayor potencia recibida, como ya se hizo previamente en [1].

Para el segundo método se realiza exactamente lo mismo que en [1], pero, al contrario, es decir, se busca un valor mínimo mediante un método redundante y se establece como un valor mínimo un valor superior al doble de la longitud del escenario (+30 metros).

```

%Obtengo las 3 Distancias menores
A=R;
min=31;
for k=1:4
    if(A(k) < min)
        min=A(k);
        MAC1=MACS(k);
        R1=min;
        i_borrar=k;
    end
end
end

```

Esto se realiza dos veces más para hallar los otros dos puntos que forman los tres puntos del triángulo.

Como conocemos las MACS y sus correspondientes valores de RSSI y Distancia para cada método, falta determinar la posición del módulo en función de su MAC, para ello simplemente se realiza un ciclo que recorra el vector que contiene las tres MACS y asigne las coordenadas de los módulos en metros en función de dicho valor

```

MACS=[MAC1 MAC2 MAC3];

for k=1:3
    if MACS(k)==1
        X(k)=7.8;
        Y(k)=13.8;
    elseif MACS(k)==2
        X(k)=0.6;
        Y(k)=0.6;
    elseif MACS(k)==4
        X(k)=0.6;
        Y(k)=11.4;
    elseif MACS(k)==5
        X(k)=8.7;
        Y(k)=0.6;
    end
end
end

```

Tras asignar las coordenadas de los módulos, hace falta introducir un pequeño valor aleatorio de ruido para evitar casos donde el divisor se acerca a 0 y se produce un error con valores infinitos en la obtención de los puntos.

A continuación, es necesario conocer para el primer caso de estudio el valor de la distancia a cada módulo, empleando para ello el método path loss que ya se empleó en [1].

Es necesario recordar que “Median” o “n” es un valor constante de calibración denominado path loss, obtenido directamente de Internet en otro estudio que emplea los mismos módulos BLE, visto en [11].

Finalmente, se obtienen las coordenadas del dispositivo móvil mediante trilateración.

```

va = ((r2*r2-r3*r3) - (x2*x2-x3*x3) - (y2*y2-y3*y3)) / 2;

vb = ((r2*r2-r1*r1) - (x2*x2-x1*x1) - (y2*y2-y1*y1)) / 2;

temp1 = vb*(x3-x2) - va*(x1-x2);

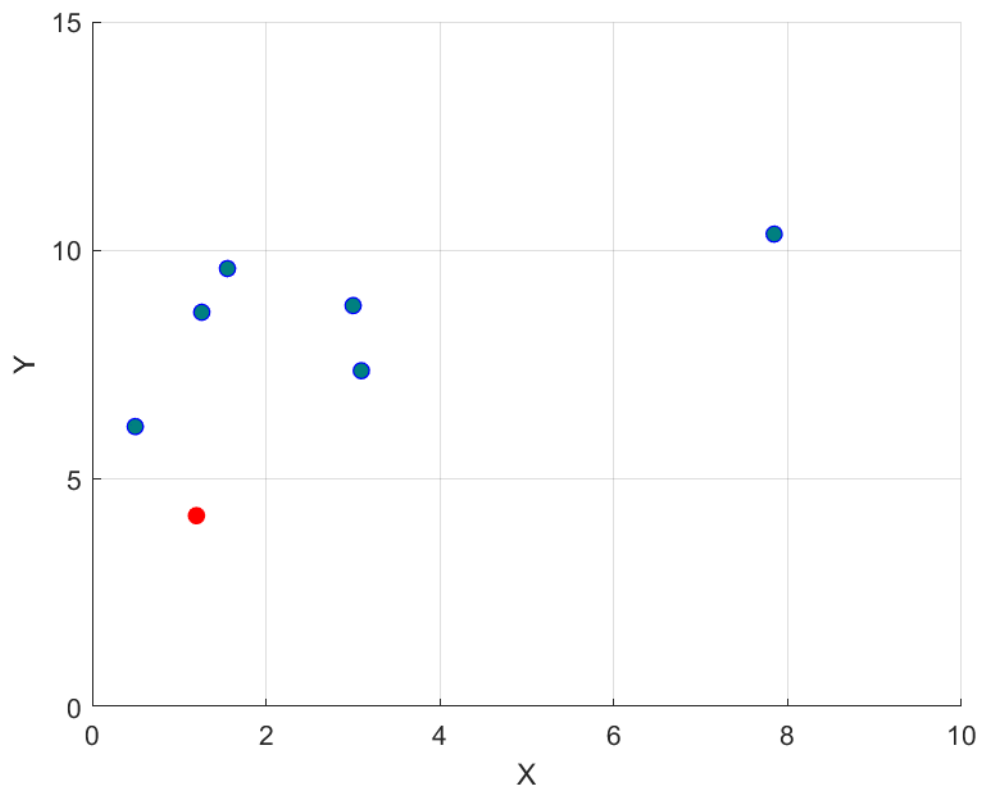
temp2 = (y1-y2)*(x3-x2) - (y3-y2)*(x1-x2);

%Posición estimada del usuario mediante trilateración:

y = temp1 / temp2;
x = (va - y*(y3-y2)) / (x3-x2);

```

Por último, sólo queda dibujar la nube de puntos obtenida para cada punto de medida, mediante la evaluación del valor de la coordenada y el valor anterior. Para ello, es necesario evaluar si el punto en el que estamos difiere del anterior y si se trata del último punto del grupo formado por las 4 medidas. En caso positivo se dibuja cada punto obtenido de color azul y de color rojo el punto donde se realiza la medida para comprobar la precisión de los resultados. Cada nube de puntos se guarda en un archivo .png con el nombre de la coordenada y se suma 4 al contador de cada iteración para pasar al siguiente cuarteto de valores.



*Figura 27: Ejemplo de la nube de puntos obtenida.*

Lo primero que choca al ver la imagen es que no hay 8 puntos como en teoría deberían aparecer. Esto sucede, principalmente por errores en la obtención de las medidas, debido a que muchos puntos obtenidos se ubican fuera de las inmediaciones del aula, es decir, en el exterior del espacio de medida definido.

#### *4.3.4 Análisis de los resultados obtenidos*

Por último, es necesario analizar los resultados obtenidos en las nubes de puntos creadas mediante el método de trilateración para cada método empleado. Al tratarse de 84 nubes, lo mejor será analizar varias muestras seleccionadas para evitar complicaciones visuales en vez de coger las 84 nubes de cada estudio.

La selección de las muestras para el estudio se ha realizado de la siguiente manera:

- Se han escogido en total 15 muestras para cada estudio, siendo cada muestra una nube de puntos correspondiente a un punto de medida.
- De las 15 muestras, 3 corresponden a muestras tomadas en el centro del escenario, permitiendo comprobar las conclusiones presentadas en [1] para ver cómo afecta a los resultados el hecho de tomar las medidas dentro del área formado por los módulos, manteniendo una distancia entre ellos de mínimo 4 metros.
- Las otras 12 muestras restantes corresponden a 3 muestras tomadas en cada esquina del escenario de medida, correspondiéndose a los puntos más cercanos a cada baliza. Esto nos permite comprobar otras de las conclusiones determinadas en [1]. Las tres muestras de la esquina superior izquierda se encuentran fuera del área encerrado por los módulos BT (recordemos que no forman un rectángulo, sino un polígono), lo que nos permite estudiar si los resultados difieren al situarse fuera del área encerrado por las balizas. Por último, al situarse en las esquinas, junto a las balizas, dichos puntos de medida se hallan pegados prácticamente a las paredes, lo que nos permite estudiar también si las paredes producen algún efecto en la reflexión/refracción y si en este trabajo sucede lo mismo que en [1], y los resultados mejoran.



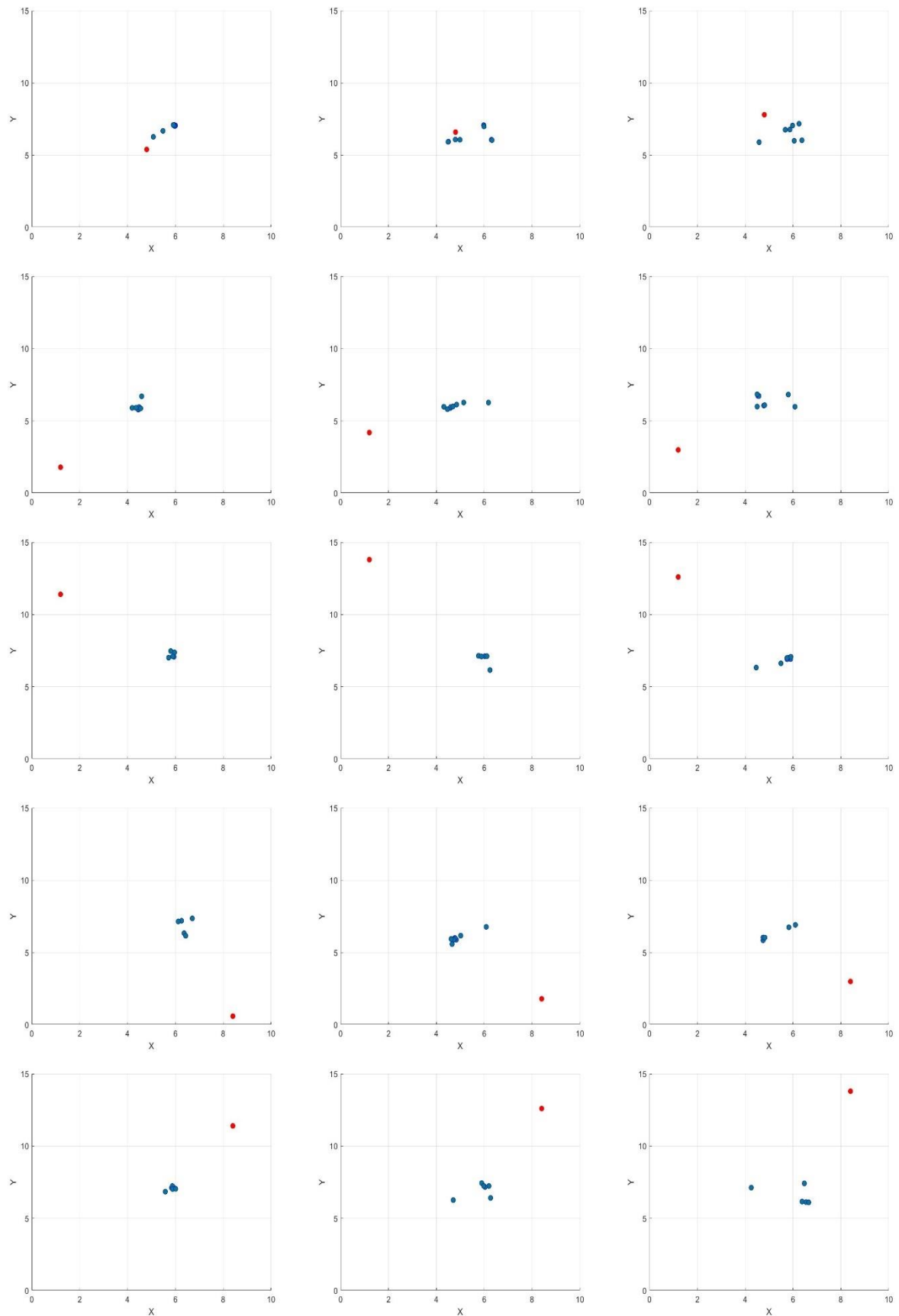


Figura 28: Nubes de puntos / Método Path Loss.

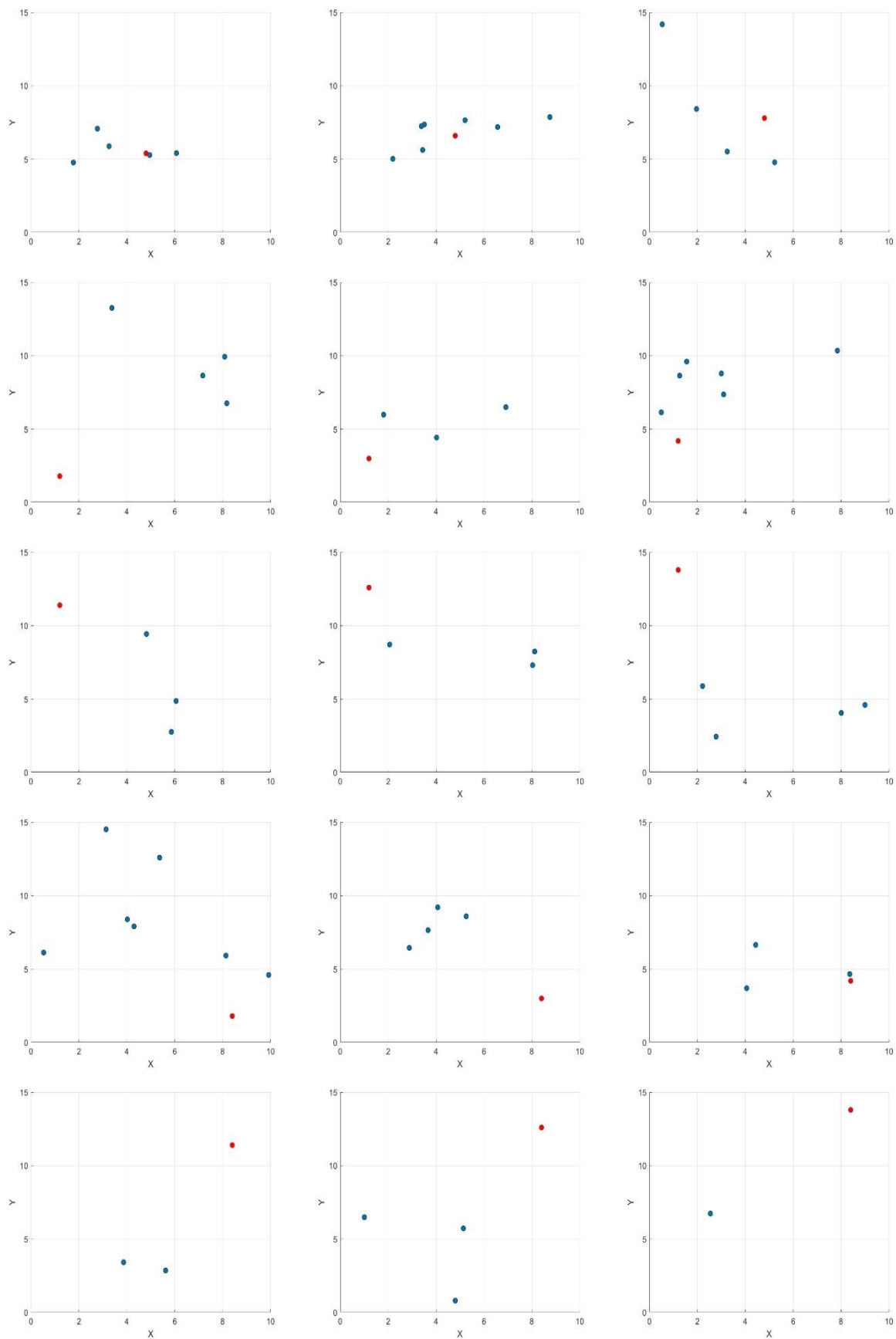


Figura 29: Nubes de puntos / Método Ranging

<i>Posición en la sala</i>	<i>Dentro del área</i>	<i>Puntos dentro</i>	<i>Puntos fuera/No generados</i>	<i>% Error en la obtención de puntos</i>
<i>Central</i>	Sí	5	3	37,5
<i>Central</i>	Sí	7	1	12,5
<i>Central</i>	Sí	7	1	12,5
<i>Inferior Izquierda</i>	Sí	7	1	12,5
<i>Inferior Izquierda</i>	Sí	8	0	0
<i>Inferior Izquierda</i>	Sí	7	1	12,5
<i>Superior Izquierda</i>	No	5	3	37,5
<i>Superior Izquierda</i>	No	5	3	37,5
<i>Superior Izquierda</i>	No	5	3	37,5
<i>Inferior Derecha</i>	Sí	5	3	37,5
<i>Inferior Derecha</i>	Sí	8	0	0
<i>Inferior Derecha</i>	Sí	6	2	25
<i>Superior Derecha</i>	Sí	5	3	37,5
<i>Superior Derecha</i>	Sí	6	2	25
<i>Superior Derecha</i>	Sí	5	3	37,5

Tabla 4: Resultados obtenidos método Path Loss

<i>Posición en la sala</i>	<i>Dentro del área</i>	<i>Puntos dentro</i>	<i>Puntos fuera/No generados</i>	<i>% Error en la obtención de puntos</i>
<i>Central</i>	Sí	5	3	37,5
<i>Central</i>	Sí	7	1	12,5
<i>Central</i>	Sí	4	4	50
<i>Inferior Izquierda</i>	Sí	4	4	50
<i>Inferior Izquierda</i>	Sí	3	5	62,5
<i>Inferior Izquierda</i>	Sí	6	2	25
<i>Superior Izquierda</i>	No	3	5	62,5
<i>Superior Izquierda</i>	No	3	5	62,5
<i>Superior Izquierda</i>	No	4	4	50
<i>Inferior Derecha</i>	Sí	7	1	12,5
<i>Inferior Derecha</i>	Sí	4	4	50
<i>Inferior Derecha</i>	Sí	3	5	62,5
<i>Superior Derecha</i>	Sí	2	6	75
<i>Superior Derecha</i>	Sí	3	5	62,5
<i>Superior Derecha</i>	Sí	1	7	87,5

Tabla 5: Resultados obtenidos método Ranging

Tras el análisis de los resultados obtenidos en la generación de puntos, se puede observar una clara ventaja a favor del primer método, obteniendo un error medio del 24,16%, mientras que en el segundo caso el error en la obtención de los puntos se dispara, situándose en un error medio del 50,83%. Es decir, que para el nuevo método se consiguen únicamente situar alrededor de 4 de cada 8 puntos dentro del escenario de medida.

A pesar de estos resultados, las nubes de puntos del segundo método se acercan más a la realidad ya que, en el primer caso, obtenemos siempre una nube de puntos situada en el centro del plano del escenario. Esto posiblemente se deba a algún error a la hora de tomar las medidas por la aplicación o a la interpretación de los datos. Es decir, aunque el porcentaje de error en la generación de puntos sea considerablemente inferior, dichos puntos se encuentran sospechosamente siempre en el centro de la sala, lo que conlleva a pensar que existe algún error en las medidas o en el procedimiento.

Hay que añadir además que las nubes de puntos generadas en el segundo método sitúan en la mayoría de los casos los puntos mucho más cerca del punto de medida que en el primer método empleado, pues no olvidemos que, como se comentó en el apartado 4.3.1, el error en la distancia del primer método se dispara hasta los 5,638 metros mientras que en el nuevo método introducido el error medio en la distancia es de 2,66 metros.

Un hecho que es común a ambos es que tanto en el primer método como en el segundo se obtiene una nube de puntos más precisa y cercana al punto de medida cuando el dispositivo móvil se sitúa en el centro de la estancia, dando resultados mucho más imprecisos en los valores más extremos de la sala, las esquinas, a diferencia de los datos obtenidos en [1].

## 5. CONCLUSIONES Y POSIBLES MEJORAS

En este último apartado se hará una breve reflexión sobre los resultados obtenidos a lo largo de este trabajo y la aportación de posibles ideas y métodos para mejorar los resultados del proyecto en una línea futura, así como entender mejor esta tecnología.

### 5.1 Conclusiones

A lo largo de este trabajo se ha presentado la incorporación de una nueva tecnología como posible alternativa de mejora al trabajo anterior para comprobar y comparar una posible mejoría en los resultados. Además, se han estudiado los posibles efectos de la orientación del dispositivo móvil en las medidas, pudiendo afirmar a través de las pruebas remitidas en este trabajo, que no existe una clara evidencia de que el cuerpo sea capaz de actuar como pantalla de forma notoria, contestando así a la conclusión planteada en el TFG anterior.

Tras analizar el algoritmo inicial, se han realizado una serie de cambios y mejoras adaptadas a este trabajo y entorno en específico, comparando los resultados obtenidos empleando dos métodos diferentes de trabajo, basados en el mismo principio del algoritmo de trilateración.

A diferencia de lo que se esperaba, no se han mejorado los resultados que se obtuvieron en el trabajo anterior. Si bien es verdad que las condiciones no han sido las mismas para dotar de riqueza el proyecto, los resultados han diferido notablemente a peor a la hora de emplear el mismo método que ya se empleó previamente para la obtención de la medida, lo que indica que necesita margen de mejora y planificación.

Sin embargo, los datos tampoco son alarmantes, ya que, a pesar de que el posicionamiento no ha resultado efectivo, los módulos que emplean la tecnología IBeacon con una distancia entre ellos de entre 10 y 15 metros (el largo de la sala) suelen tener una precisión que ronda los 4 metros, a diferencia de los 5,6 obtenidos en este trabajo.

La verdadera sorpresa en este trabajo han sido los resultados obtenidos en el error cometido en la medida empleando el segundo método, situándose en 2,66 metros de precisión, a diferencia de los 4 metros de media generales y los 5,6 obtenidos en el primer método, lo que nos lleva a pensar que, introduciendo una serie de posibles mejoras y metodologías a este trabajo en un futuro, se pueda emplear esta tecnología partiendo de este método.

Desde el punto de vista técnico, este trabajo ha demostrado la viabilidad de realizar despliegues de balizas BT sin necesidad de infraestructuras adicionales. Únicamente empleando módulos BLE económicos (no olvidemos que su precio es inferior a 1 euro) y baterías LiPo, se ha conseguido determinar de forma sorprendentemente rápida la

posición del dispositivo móvil, aunque no de una forma totalmente precisa, como era de esperar.

## 5.2 Posibles mejoras en el futuro

Pese a que los resultados del segundo método empleado son algo esperanzadores, este trabajo suscita muchas dudas en el aire ya que los primeros resultados son considerablemente peores y abre la puerta a líneas futuras de mejora mediante aplicaciones de nuevas metodologías de trabajo, cambios en el entorno o los BLE empleados, o en el uso y desarrollo de las aplicaciones.

- **Mejoras de los materiales:** Como posible mejora inicial, podría ser un acierto incorporar nuevos BLE que porten una versión de bluetooth superior a los HM-10 y que se acerque a la versión actual de Bluetooth, 5.x. Esto supondría mejoras en la velocidad, mayor fiabilidad y rango de cobertura, produciéndose así una evolución realmente favorable en el aspecto técnico fundamental del trabajo. Esto también beneficiaría al estudio en términos de robustez frente a posibles interferencias, otra de las posibles conclusiones en el fallo de la toma de las medidas en este trabajo.
- **Mejoras de la aplicación de medida:** Si bien emplear un entorno de desarrollo de aplicaciones sandbox ha facilitado la realización de este trabajo, lo ideal hubiese sido crear una aplicación desde 0 que hubiese trabajado y estudiado el comportamiento de los módulos desde el interior de estos, entendiendo a la perfección su funcionamiento de trabajo y jugando con ello a nuestro favor mediante el empleo de analizadores de tramas e información como WireShark.
- **Mejoras de metodología de trabajo:** La obtención de las medidas mediante una aplicación móvil, que requiere directamente del estado del hardware y software de este, y posteriormente tener que pasar todos estos datos mediante su exportación a una plataforma de trabajo con hojas de cálculo como Excel hace que, inevitablemente, se puedan cometer errores en los datos a la hora de importarlos u obtenerlos directamente en la medida. Otro de las posibles causas de error podría haber sido el entorno de trabajo, dado que el aula se sitúa en una de las plantas inferiores de la universidad, expuesta a toda clase de interferencias de diferentes tipos de tecnología, desde smartphones, pasando por routers WiFi, hasta antenas de telecomunicaciones.

## 6. BIBLIOGRAFÍA

- [1] Calzada Cabano, Carlos. Análisis y mejora de un algoritmo de posicionamiento en interiores. 10-2018. Disponible en: <https://repositorio.unican.es/xmlui/handle/10902/15537>
- [2] Stefanov Ivanov, Rosen. Sensors: An Indoor Location-Based Control System Using Bluetooth Beacons for IoT Systems. Disponible en: <https://fags.tips/post/i-want-matlab-code-for-trilateration-algorithmcan-any-one-help-me.html>
- [3] Pérez, Antonio. Posicionamiento en interiores (indoor positioning). Disponible en: <https://informatica.blogs.uoc.edu/posicionamiento-en-interiores-indoor-positioning/>
- [4] Iglesias, Pablo. Sistemas de posicionamiento en interiores. Disponible en: <https://www.pablovglesias.com/sistemas-de-posicionamiento-en-interiores/>
- [5] Una guía detallada de IBeacon. Disponible en: <https://www.mokoblue.com/es/a-detailed-guide-to-ibeacon/>
- [6] Grzechca, Damian. Analysis of Object Location Accuracy for iBeacon Technology based on the RSSI Path Loss Model and Fingerprint Map. Disponible en: [https://www.researchgate.net/publication/311341399\\_Analysis\\_of\\_Object\\_Location\\_Accuracy\\_for\\_iBeacon\\_Technology\\_based\\_on\\_the\\_RSSI\\_Path\\_Loss\\_Model\\_and\\_Fingerprint\\_Map](https://www.researchgate.net/publication/311341399_Analysis_of_Object_Location_Accuracy_for_iBeacon_Technology_based_on_the_RSSI_Path_Loss_Model_and_Fingerprint_Map)
- [7] Using HM-10 BLE Modules as Low-Cost IBeacons. Disponible en: <http://www.blueluminance.com/HM-10-as-iBeacon.pdf>
- [8] TracerPlus Software. Disponible en: <https://www.tracerplus.com/software/downloads>
- [9] HM-10 Datasheet. Disponible en: [http://www.jnhuamao.cn/bluetooth40\\_en.zip](http://www.jnhuamao.cn/bluetooth40_en.zip)
- [10] Villalpando, Juan Antonio. Tutorial del Internet de las Cosas y Bluetooth con el HM10. Disponible en: [http://kio4.com/arduino/161\\_HM10\\_BLE\\_Actualizar.htm](http://kio4.com/arduino/161_HM10_BLE_Actualizar.htm)